用户手册

NEC

# 78K0/KC2

## 8 位单片微控制器

---

**μPD78F0511　μPD78F0511(A)**
**μPD78F0512　μPD78F0512(A)**
**μPD78F0513　μPD78F0513(A)**
**μPD78F0514　μPD78F0514(A)**
**μPD78F0515　μPD78F0515(A)**
**μPD78F0513D**
**μPD78F0515D**

μPD78F0513D 和 78F0515D 具有片上调试功能。
大规模生产时不要使用本产品，因为 flash 存储器可重写的次数是一定的，在使用片上调试功能后，不能保证它的可靠性。
NEC 电子还没有接到关于此产品的投诉。

文档编号　U17336EJ4V0UD00 (第四版)
发行日期　March 2006 NS　CP(K)

[备忘录]

## CMOS 设备注意事项

### ① 输入引脚处的电压适用波形

因输入噪声或反射波产生的波形失真可能产生故障。CMOS 设备的输入端电平停留在 VIL(最大)和 VIH(最小)之间，该设备可能产生故障。当输入电平固定时，和在传输阶段当输入电平在 VIL(最大)和 VIH(最小)之间时，切记避免波动噪声进入设备。

### ② 未使用的 CMOS 输入管脚的处理

CMOS 设备输入端未连接将会产生工作故障。如果输入管脚未连接，则由于噪音等原因可能会产生内部输入电平，从而导致故障。CMOS 设备的运行与 Bipolar 或 NMOS 设备不同。必须借助上拉或下拉电路使 CMOS 设备的输入电平固定在高电平或是低电平。如果每个未用的管脚可作为输出管脚的话，那么它们可以通过电阻与 VDD 或 GND 连接。对未使用管脚的处理因设备而异，必须遵循与设备相关的规定和说明。

### ③ 半导体的 ESD 防护措施

如果 MOS 设备周围有强电场，将会击穿氧化栅极，从而影响设备的运行。因此必须采取措施，尽可能防止静电产生，而且一旦有静电，必须立即释放。对于环境必须有适当的控制。如果空气干燥，应当使用增湿器。建议避免使用绝缘体，因为它们容易产生静电。半导体设备的存放和运输必须在抗静电容器、抗静电屏蔽袋或导电材料容器中进行。所有的测试和测量工具包括工作台和工作面必须良好接地。操作员应当佩戴静电消除手套以保证良好接地，不能用手直接接触半导体设备。对于装配有半导体设备的 PW 板也应采取类似的静电防范措施。

### ④ MOS 设备初始化之前的状态

上电并不能确定 MOS 设备的初始状态。在刚刚上电之后，具有复位功能的 MOS 设备并没有被初始化，因此上电不能保证输出管脚的电平，I/O 设置和寄存器的内容。设备在收到复位信号后才进行初始化，具有复位功能的设备在上电后必须立即进行复位操作。

### ⑤ 上/下电顺序

在器件内部操作和外部接口使用不同供电电源的情况下，上电顺序的规则是在内部电源打开后再打开外部供电电源。下电的规则是先关掉外部供电电源，然后是内部电源。使用如上相反的顺序上下电可能导致器件内部元件过压，从而导致故障和因为不正常的电流致使内部元件退化。
正确的上下电顺序因设备而异，必须遵循与设备相关的规定和说明。

### ⑥ 下电状态的信号输入

在器件没上电时，不要输入信号或者一个 I/O 上拉电源。因输入该信号或 I/O 上拉电源产生的电流注射可能产生故障和产生流过器件的不寻常的电流，从而在那时产生内部元件退化。在下电状态下的信号输入因设备而异，必须遵循与设备相关的规定和说明。

**EEPROM** 是 **NEC** 电子公司的注册商标**.**
**Windows** 和 **Windows NT** 是美国及其它国家微软公司的注册商标**.**
**PC/AT** 是 **IBM** 公司的注册商标**.**
**HP9000** 系列 **700** 和 **HP-UX** 是 **HP** 公司的注册商标**.**
**SPARC** 工作站是 **SPARC** 公司的注册商标**.**
**Solaris** 和 **SunOS** 是 **Sun** 公司的注册商标**.**

**SuperFlash 是 Silicon Storage Technology,Inc(SST 超捷) 的注册商标,在包括美国日本的多个国家注册.**

# 引 言

**读者对象**　　本手册于适用那些希望了解 78K0/KC2 功能，并设计开发应用系统和程序的工程师。
目标产品如下。

78K0/KC2: μPD78F0511, 78F0512, 78F0513, 78F0514, 78F0515, 78F0513D,
<R> 78F0515D, 78F0511(A), 78F0512(A), 78F0513(A), 78F0514(A),
78F0515(A)

**目 的**　　本手册用于帮助用户了解下面组件中描述的功能。

**组 件**　　78K0/KC2 手册主要分为两个部分：手册和指令（与 78K/0 系列通用）。

| **78K0/KC2**<br>用户手册<br>**(本手册)** | **78K/0 系列**<br>用户手册<br>指令 |
|---|---|
| • 引脚功能<br>• 内部模块功能<br>• 中断<br>• 其它内置外设功能<br>• 电气特性 | • CPU 功能<br>• 指令集<br>• 指令扩展 |

**手册使用方法**　　在阅读本手册前，读者应掌握电子工程、逻辑电路和微控制器等方面的一般知识。

<R>
- 当针对 (A) 级产品使用本手册时：
  → 标准产品与(A) 级产品仅仅是品质等级的差异。请查看以下对应。
  - μPD78F0511→ μPD78F0511(A)
  - μPD78F0512→ μPD78F0512(A)
  - μPD78F0513→ μPD78F0513(A)
  - μPD78F0514→ μPD78F0514(A)
  - μPD78F0515→ μPD78F0515(A)
- 如果读者要了解产品功能:
  → 请按**目录**顺序阅读本手册。标识 “<R>” 处是主要修改的地方。在 PDF 文件中可以使用查找 “<R>”，来很容易的找到修改的地方。
- 如何解释寄存器格式：
  → 括号中的位名在 RA78K0 中被定义为保留字，并且在 CC78K0 中直接通过 # pragma sfr 定义为一个 sfr 变量。
- 如果读者知道寄存器名字并想验证这个寄存器的详细信息：
  → 可参阅**附录 C 寄存器索引**。

- 如果读者希望了解 78K/0 系列指令的详细信息：
  → 可参阅 **78K/0 系列指令用户手册（U12326E）**。

**规 定**

| | | |
|---|---|---|
| 数据规则: | 数据的高位部分在左边，低位部分在右边 | |
| 有效低电平表示法: | $\overline{\times\times\times}$ (在引脚和信号名称上划一条线) | |
| **注**: | 文中用**注**标注的相关术语的脚注 | |
| **注意事项**: | 需要特别关注的信息 | |
| **备注**: | 补充信息 | |
| 数的表示法: | 二进制 | …×××× 或 ××××B |
| | 十进制 | …×××× |
| | 十六进制 | …××××H |

**相关文档** 本手册中指出的相关文档可能包括了初级的版本，但未注明。

**相关设备文档**

| | 文档名称 | 文档编号 |
|---|---|---|
| | 78K0/KC2 用户手册 | 本手册 |
| | 78K/0 系列指令用户手册 | U12326E |
| <R> | 78K0/Kx2 Flash 存储器编程 (编程器) 应用注释 | U17739E |
| <R> | 78K0/Kx2 Flash 存储器自编程用户手册 [注] | U17516E |

**注** 本手册是针对技术而做的说明。详细问题，请咨询 **NEC** 电子的销售代表。

**开发工具相关文档 (软件)(用户手册)**

| 文档名称 | | 文档编号 |
|---|---|---|
| RA78K0 Ver. 3.80 汇编包 | 操作 | U17199E |
| | 语言 | U17198E |
| | 结构化汇编语言 | U17197E |
| CC78K0 Ver. 3.70 C 编译器 | 操作 | U17201E |
| | 语言 | U17200E |
| SM+ 系统仿真器 | 操作 | U17246E |
| | 外部用户开放接口 | U17247E |
| ID78K0-QB Ver. 2.90 集成调试器 | 操作 | U17437E |
| PM+ Ver. 5.20 | | U16934E |

**开发工具相关文档 (硬件)(用户手册)**

| 文档名称 | 文档编号 |
|---|---|
| QB-78K0KX2 在线仿真器 | U17341E |
| QB-78K0MINI 片上调试仿真器 | U17029E |

**注意事项** 对以上列出的相关文档所做修改恕不另行通知，在设计时请使用每个文档的最新版本。

与 **Flash** 存储器编程相关文档 **(**用户手册**)**

| | 文档名称 | 文档编号 |
|---|---|---|
| | PG-FP4 Flash 存储器编程器 | U15260E |
| <R> | PG-FPL3 Flash 存储器编程器 | U17454E |

其它文档

| 文档名称 | 文档编号 |
|---|---|
| 半导体选择指南－产品和封装－ | X13769X |
| 半导体设备安装手册 | 注 |
| NEC 半导体设备质量等级 | C11531E |
| NEC 半导体设备可靠性/质量控制系统 | C10983E |
| 半导体设备防静电 ESD 保护指南 | C11892E |

注　　浏览“半导体设备安装手册”网站 ( http://www.necel.com/pkg/en/mount/index.html ).

注意事项　对以上列出的相关文档所做修改恕不另行通知，在设计时请使用每个文档的最新版本。

目 录

# 第一章　概述

## 1.1 特征

- 指令最短执行时间可以在高速(0.1 $\mu$s: @ 高速系统时钟的操作频率为 20 MHz) 和超低速(122 $\mu$s: @副系统时钟的操作频率为 32.768 kHz)之间改变
- 通用寄存器: 8 位 × 32 个寄存器(8 位× 8 个寄存器× 4 组)
- ROM, RAM 的容量

| 项目<br>产品型号 | 程序存储器<br>(ROM) | | 数据存储器 | |
|---|---|---|---|---|
| | | | 内部高速 RAM[注] | 内部扩展 RAM[注] |
| $\mu$PD78F0511 | Flash 存储器[注] | 16 KB | 768 字节 | – |
| $\mu$PD78F0512 | | 24 KB | 1 KB | |
| $\mu$PD78F0513, 78F0513D | | 32 KB | | |
| $\mu$PD78F0514 | | 48 KB | | 1 KB |
| $\mu$PD78F0515, 78F0515D | | 60 KB | | 2 KB |

注　通过使用内存容量切换寄存器(IMS)和内部扩展 RAM 容量切换寄存器(IXS)，可以改变内部 Flash 存储器，内部高速 RAM 容量和内部扩展 RAM 的容量。要了解 IMS 和 IXS，请参考 **25.1 内存容量切换寄存器** 和 **25.2 内部扩展 RAM 容量切换寄存器**。

- 内置单电源 Flash 存储器
- 自编程(具有启动交换功能)
- 片上调试功能(仅用于$\mu$PD78F0513D 和 78F0515D )[注]
- 内置上电清零 (POC) 电路和低电压检测器(LVI)
- 内置看门狗定时器(使用内置的内部低速振荡时钟进行操作)
- 内置乘法器/除法器 (16 位 × 16 位, 32 位/ 16 位)
  (仅用于$\mu$PD78F0514, 78F0515, 和 78F0515D )
- 内置按键中断功能
- 内置时钟输出控制器
- I/O 端口：
  44 引脚产品: 37 (含 N 沟开漏: 4)
  48 引脚产品: 41 (含 N 沟开漏: 4)

注　$\mu$PD78F0513D 和 78F0515D 具有片上调试功能。从重复写入 Flash 存储器的次数有限的观点看，在使用片上调试功能后它的可靠性并不能得到保证，所以使用此产品不要用于大规模生产。NEC 电子不接受对于本产品的任何投诉 。

- 定时器: 7 通道
  - 16 位定时器/事件计数器:　1 通道
  - 8 位定时器/事件计数器:　2 通道
  - 8 位定时器:　2 通道
  - 钟表定时器:　1 通道
  - 看门狗定时器:　1 通道
- 串行接口: 3 通道
  - UART 支持 LIN (本地互联网络)- 总线:　1 通道
  - CSI/UART[注]:　1 通道
  - $I^2C$:　1 通道
- 10 位分辨率 A/D 转换 ($AV_{REF}$ = 2.3 到 5.5 V): 8 通道
- 电源电压: $V_{DD}$ = 1.8 到 5.5 V
- <R> 工作环境温度: $T_A$ = –40 到 +85°C

**注**　选择这些复用功能引脚中的任意一种功能。

## 1.2 应 用

- 汽车设备 ( (A), (A1), (A2) 级产品，正在开发中 )
  - 车身电子系统控制(电动门窗，车门开关接收等)
  - 用于控制的次级微控制器
- 汽车音响
- AV 设备，家用音响
- PC 外围设备 (键盘等)
- 家用电器
  - 空调
  - 微波炉，电饭煲
- 工业设备
  - 电机
  - 自动售货机
  - FA (工业自动化)

<R>

## 1.3 订购信息

- **Flash 存储器版本 (1/4)**

| 型　号 | 封　装 | 等　级 |
|---|---|---|
| PD78F0511GA-8EU-A | 48 引脚塑封 LQFP (密脚距) (7x7) | 标　准 |
| PD78F0511GB-UES-A | 44 引脚塑封 LQFP (10x10) | 标　准 |
| PD78F0512GA-8EU-A | 48 引脚塑封 LQFP (密脚距) (7x7) | 标　准 |
| PD78F0512GB-UES-A | 44 引脚塑封 LQFP (10x10) | 标　准 |
| PD78F0513GA-8EU-A | 48 引脚塑封 LQFP (密脚距) (7x7) | 标　准 |
| PD78F0513GB-UES-A | 44 引脚塑封 LQFP (10x10) | 标　准 |
| PD78F0514GA-8EU-A | 48 引脚塑封 LQFP (密脚距) (7x7) | 标　准 |
| PD78F0515GA-8EU-A | 48 引脚塑封 LQFP (密脚距) (7x7) | 标　准 |
| PD78F0513DGB-UES-A[注1] | 44 引脚塑封 LQFP (10x10) | 标　准 |
| PD78F0515DGA-8EU-A[注1] | 48 引脚塑封 LQFP (密脚距) (7x7) | 标　准 |
| PD78F0511GA(A)-GAM-AX[注2] | 48 引脚塑封 LQFP (密脚距) (7x7) | 专　业 |
| PD78F0511GB(A)-GAF-AX[注2] | 44 引脚塑封 LQFP (10x10) | 专　业 |
| PD78F0512GA(A)-GAM-AX[注2] | 48 引脚塑封 LQFP (密脚距) (7x7) | 专　业 |
| PD78F0512GB(A)-GAF-AX[注2] | 44 引脚塑封 LQFP (10x10) | 专　业 |
| PD78F0513GA(A)-GAM-AX[注2] | 48 引脚塑封 LQFP (密脚距) (7x7) | 专　业 |
| PD78F0513GB(A)-GAF-AX[注2] | 44 引脚塑封 LQFP (10x10) | 专　业 |
| PD78F0514GA(A)-GAM-AX[注2] | 48 引脚塑封 LQFP (密脚距) (7x7) | 专　业 |
| PD78F0515GA(A)-GAM-AX[注2] | 48 引脚塑封 LQFP (密脚距) (7x7) | 专　业 |

注　**1.** μPD78F0513D 和 78F0515D 具有片上调试功能。从重复写入 Flash 存储器的次数有限的观点看，在使用片上调试功能后它的可靠性并不能得到保证，所以使用此产品不要用于大规模生产。NEC 电子不接受对于本产品的任何投诉 。

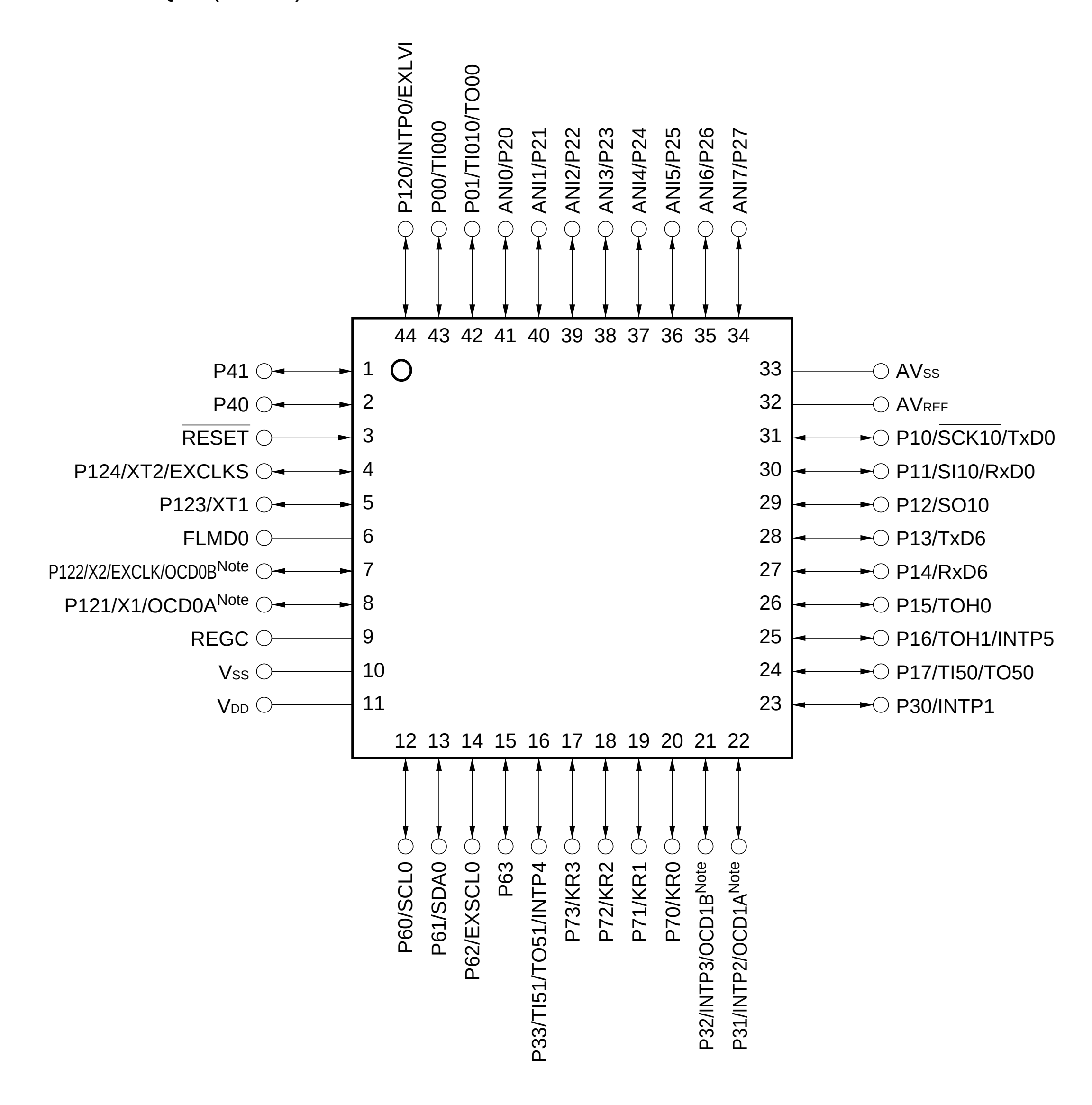
　　**2.** 开发中。

**备注**　在产品型号末尾有 **-A** 和 **-AX** 的是无铅产品。

请参考 NEC 电子出版的“NEC 半导体设备质量等级”(文档编号 C11531E)，查看设备质量等级的说明和推荐的应用领域。

## 1.4 引脚配置(俯视图)

- 44 引脚塑封 LQFP (10 × 10)




注　仅适用于μPD78F0513D (具有片上调试功能的产品)

注意事项　**1.** 使 **AVss** 与 **Vss** 的电位相同。

<R>　**2.** 通过一个电容**(0.47μF** 到 **1μF:** 推荐**)**将 **REGC** 引脚和 **Vss** 连接到一起。

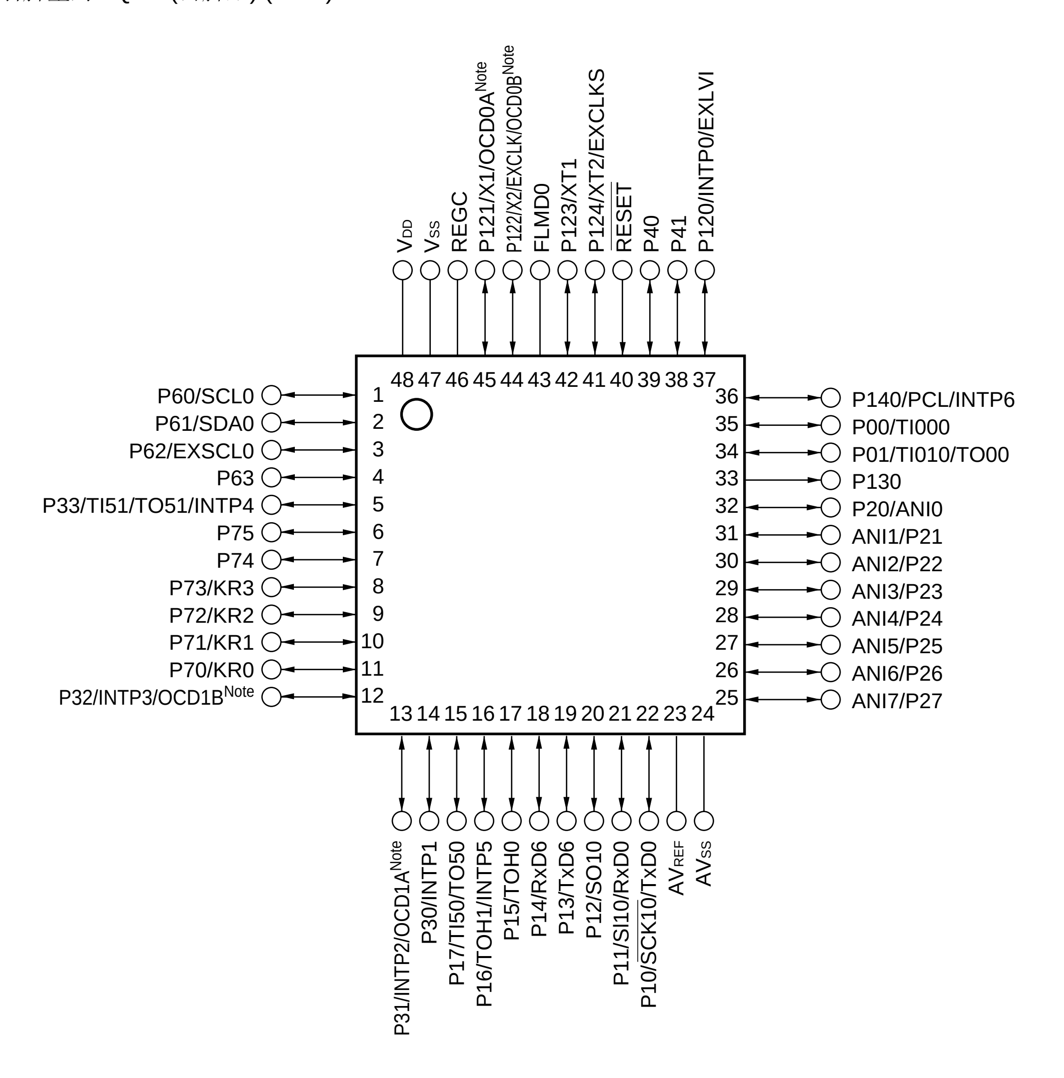
**3.** 复位后，**ANI0/P20** 到 **ANI7/P27** 设置为模拟输入模式。

引脚标识

| 引脚 | 功能 |
|---|---|
| ANI0 ～ ANI7: | 模拟输入 |
| $AV_{REF}$: | 模拟参考电压 |
| $AV_{SS}$: | 模拟地 |
| EXCLK: | 外部时钟输入<br>(主系统时钟) |
| EXCLKS: | 外部时钟输入<br>(副时钟) |
| EXLVI: | 外部电压输入<br>用于低电压检测 |
| EXSCL0: | 外部串行时钟输入 |
| FLMD0: | Flash 编程模式 |
| INTP0 ～ INTP5: | 外部中断输入 |
| KR0 ～ KR3: | 按键返回 |
| OCD0A[注],<br>OCD0B[注],<br>OCD1A[注],<br>OCD1B[注]: | 片上调试输入/输出 |
| P00 , P01: | 端口 0 |
| P10 ～ P17: | 端口 1 |
| P20 ～ P27: | 端口 2 |
| P30 ～ P33: | 端口 3 |
| P40, P41: | 端口 4 |
| P60 ～ P63: | 端口 6 |
| P70 ～ P73: | 端口 7 |
| P120 ～ P124: | 端口 12 |
| REGC | 调校电容 |
| $\overline{RESET}$: | 复位 |
| RxD0, RxD6: | 接收数据 |
| $\overline{SCK10}$, SCL0: | 串行时钟输入/输出 |
| SDA0: | 串行数据输入/输出 |
| SI10: | 串行数据输入 |
| SO10: | 串行数据输出 |
| TI000, TI010,<br>TI001, TI011,<br>TI50, TI51: | 定时器输入 |
| TO00, TO01 ,<br>TO50, TO51,<br>TOH0, TOH1: | 定时器输出 |
| TxD0, TxD6: | 发送数据 |
| $V_{DD}$: | 电源电压 |
| $V_{SS}$: | 地 |
| X1, X2: | 晶体振荡器 (主系统时钟) |
| XT1, XT2: | 晶体振荡器(副时钟) |

注　仅适用于μPD78F0513D (具有片上调试功能的产品)

- 48 引脚塑封 LQFP (密脚距) (7 × 7)




注 仅适用于μPD78F0515D (具有片上调试功能的产品)

注意事项 **1.** 使 **AVss** 与 **Vss** 的电位相同。

<R> **2.** 通过一个电容**(0.47μF** 到 **1μF:** 推荐**)**将 **REGC** 引脚和 **Vss** 连接到一起。

**3.** 复位后，**ANI0/P20** 到 **ANI7/P27** 设置为模拟输入模式。

引脚标识

| 引脚 | 说明 |
|---|---|
| ANI0 ～ ANI7: | 模拟输入 |
| $AV_{REF}$: | 模拟参考电压 |
| $AV_{SS}$: | 模拟地 |
| EXCLK: | 外部时钟输入 (主系统时钟) |
| EXCLKS: | 外部时钟输入 (副时钟) |
| EXLVI: | 外部电压输入 用于低电压检测 |
| EXSCL0: | 外部串行时钟输入 |
| FLMD0: | Flash 编程模式 |
| INTP0 ～ INTP6: | 外部中断输入 |
| KR0 ～ KR3: | 按键返回 |
| OCD0A[注], OCD0B[注], OCD1A[注], OCD1B[注]: | 片上调试输入/输出 |
| P00, P01: | 端口 0 |
| P10 ～ P17: | 端口 1 |
| P20 ～ P27: | 端口 2 |
| P30 ～ P33: | 端口 3 |
| P40, P41: | 端口 4 |
| P60 ～ P63: | 端口 6 |
| P70 ～ P75: | 端口 7 |
| P120 ～ P124: | 端口 12 |
| P130: | 端口 13 |
| P140: | 端口 14 |
| PCL: | 可编程时钟输出 |
| REGC | 调校电容 |
| $\overline{\text{RESET}}$: | 复位 |
| RxD0, RxD6: | 接收数据 |
| $\overline{\text{SCK10}}$, SCL0: | 串行时钟输入/输出 |
| SDA0: | 串行数据输入/输出 |
| SI10: | 串行数据输入 |
| SO10: | 串行数据输出 |
| TI000, TI010, TI50, TI51: | 定时器输入 |
| TO00, TO50, TO51, TOH0, TOH1: | 定时器输出 |
| TxD0, TxD6: | 发送数据 |
| $V_{DD}$: | 电源电压 |
| $V_{SS}$: | 地 |
| X1, X2: | 晶体振荡器 (主系统时钟) |
| XT1, XT2: | 晶体振荡器 (副时钟) |

注　仅适用于μPD78F0515D (具有片上调试功能的产品)

## 1.5 78K0/Kx2 系列介绍

| ROM | RAM | 78K0/KB2 | 78K0/KC2 | | 78K0/KD2 | 78K0/KE2 | 78K0/KF2 |
|---|---|---|---|---|---|---|---|
| | | 30/36 引脚 | 44 引脚 | 48 引脚 | 52 引脚 | 64 引脚 | 80 引脚 |
| 128 KB | 7 KB | – | – | – | μPD78F0527D[注] | μPD78F0537D[注] | μPD78F0547D[注] |
| | | | | | μPD78F0527 | μPD78F0537 | μPD78F0547 |
| 96 KB | 5 KB | – | – | – | μPD78F0526 | μPD78F0536 | μPD78F0546 |
| 60 KB | 3 KB | – | – | μPD78F0515D[注] | μPD78F0525 | μPD78F0535 | μPD78F0545 |
| | | | | μPD78F0515 | | | |
| 48 KB | 2 KB | – | – | μPD78F0514 | μPD78F0524 | μPD78F0534 | μPD78F0544 |
| 32 KB | 1 KB | μPD78F0503D[注] | μPD78F0513D[注] | μPD78F0513 | μPD78F0523 | μPD78F0533 | – |
| | | μPD78F0503 | μPD78F0513 | | | | |
| 24 KB | 1 KB | μPD78F0502 | μPD78F0512 | | μPD78F0522 | μPD78F0532 | – |
| 16 KB | 768 B | μPD78F0501 | μPD78F0511 | | μPD78F0521 | μPD78F0531 | – |
| 8 KB | 512 B | μPD78F0500 | – | | – | – | – |

注　　该产品具有片上调试功能

78K0/Kx2 功能列表如下：

(1/2)

| 项目 \ 型号 | 78K0/KB2 | | | | 78K0/KC2 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 30/36 引脚 | | | | 44 引脚 | | | 48 引脚 | | | | |
| Flash 存储器 (KB) | 8 | 16 | 24 | 32 | 16 | 24 | 32 | 16 | 24 | 32 | 48 | 60 |
| RAM (KB) | 0.5 | 0.75 | 1 | 1 | 0.75 | 1 | 1 | 0.75 | 1 | 1 | 2 | 3 |
| Bank (flash 存储器) | – | | | | | | | | | | | |
| 电源电压 | $V_{DD}$ = 1.8 ～ 5.5 V | | | | | | | | | | | |
| 稳压器 | 提供 | | | | | | | | | | | |
| 最小指令执行时间 | 0.1 $\mu$s (20 MHz: $V_{DD}$ = 4.0 ～ 5.5 V)/0.2 $\mu$s (10 MHz: $V_{DD}$ = 2.7 ～ 5.5 V)/ 0.4 $\mu$s (5 MHz: $V_{DD}$ = 1.8 ～ 5.5 V) | | | | | | | | | | | |
| 时钟 / 主系统时钟 / 高速系统 | 20 MHz: $V_{DD}$ = 4.0 ～ 5.5 V/10 MHz: $V_{DD}$ = 2.7 ～ 5.5 V/5 MHz: $V_{DD}$ = 1.8 ～ 5.5 V | | | | | | | | | | | |
| 时钟 / 主系统时钟 / 内部高速振荡器 | 8 MHz (TYP.): $V_{DD}$ = 1.8 ～ 5.5 V | | | | | | | | | | | |
| 时钟 / 副时钟 | – | | | | 32.768 kHz (TYP.): $V_{DD}$ = 1.8 ～ 5.5 V | | | | | | | |
| 时钟 / 内部低速振荡器 | 240 kHz (TYP.): $V_{DD}$ = 1.8 ～ 5.5 V | | | | | | | | | | | |
| 端口 / 合计 | 23 | | | | 37 | | | 41 | | | | |
| 端口 / N-ch O.D. (6 V 耐压) | 2 | | | | 4 | | | 4 | | | | |
| 定时器 / 16 位 (TM0) | 1 通道 | | | | | | | | | | | |
| 定时器 / 8 位(TM5) | 2 通道 | | | | | | | | | | | |
| 定时器 / 8 位(TMH) | 2 通道 | | | | | | | | | | | |
| 定时器 / 钟表 | – | | | | 1 通道 | | | | | | | |
| 定时器 / WDT 看门狗 | 1 通道 | | | | | | | | | | | |
| 串行接口 / 3 线 CSI | – | | | | | | | | | | | |
| 串行接口 / 自动收发的 3 线 CSI | – | | | | | | | | | | | |
| 串行接口 / UART/3 线CSI[注] | 1 通道 | | | | | | | | | | | |
| 串行接口 / 支持 LIN 总线的 UART | 1 通道 | | | | | | | | | | | |
| 串行接口 / $I^2C$ 总线 | 1 通道 | | | | | | | | | | | |
| 10 位 A/D | 4 通道 | | | | 8 通道 | | | | | | | |
| 中断 / 外部 | 6 | | | | 7 | | | 8 | | | | |
| 中断 / 内部 | 14 | | | | 16 | | | | | | | |
| 按键中断 | – | | | | 4 通道 | | | | | | | |
| 复位 / $\overline{\text{RESET}}$ 引脚 | 提供 | | | | | | | | | | | |
| 复位 / POC （上电清零） | 1.59 V ±0.15 V (电压上升到 1.8 V 所用时间: 3.6 ms (最小)) | | | | | | | | | | | |
| 复位 / LVI （低电压检测） | 可选择 16 个电源电压检测级别 | | | | | | | | | | | |
| 复位 / WDT （看门狗定时器） | 提供 | | | | | | | | | | | |
| 时钟输出/蜂鸣器输出 | – | | | | | | | 仅有时钟输出 | | | | |
| 乘法器/除法器 | – | | | | | | | | | | | 提供 |
| 片上调试功能 | 仅$\mu$PD78F0503D | | | | 仅$\mu$PD78F0513D | | | 仅$\mu$PD78F0515D | | | | |
| 工作环境温度 | $T_A$ = –40 ～ +85°C | | | | | | | | | | | |

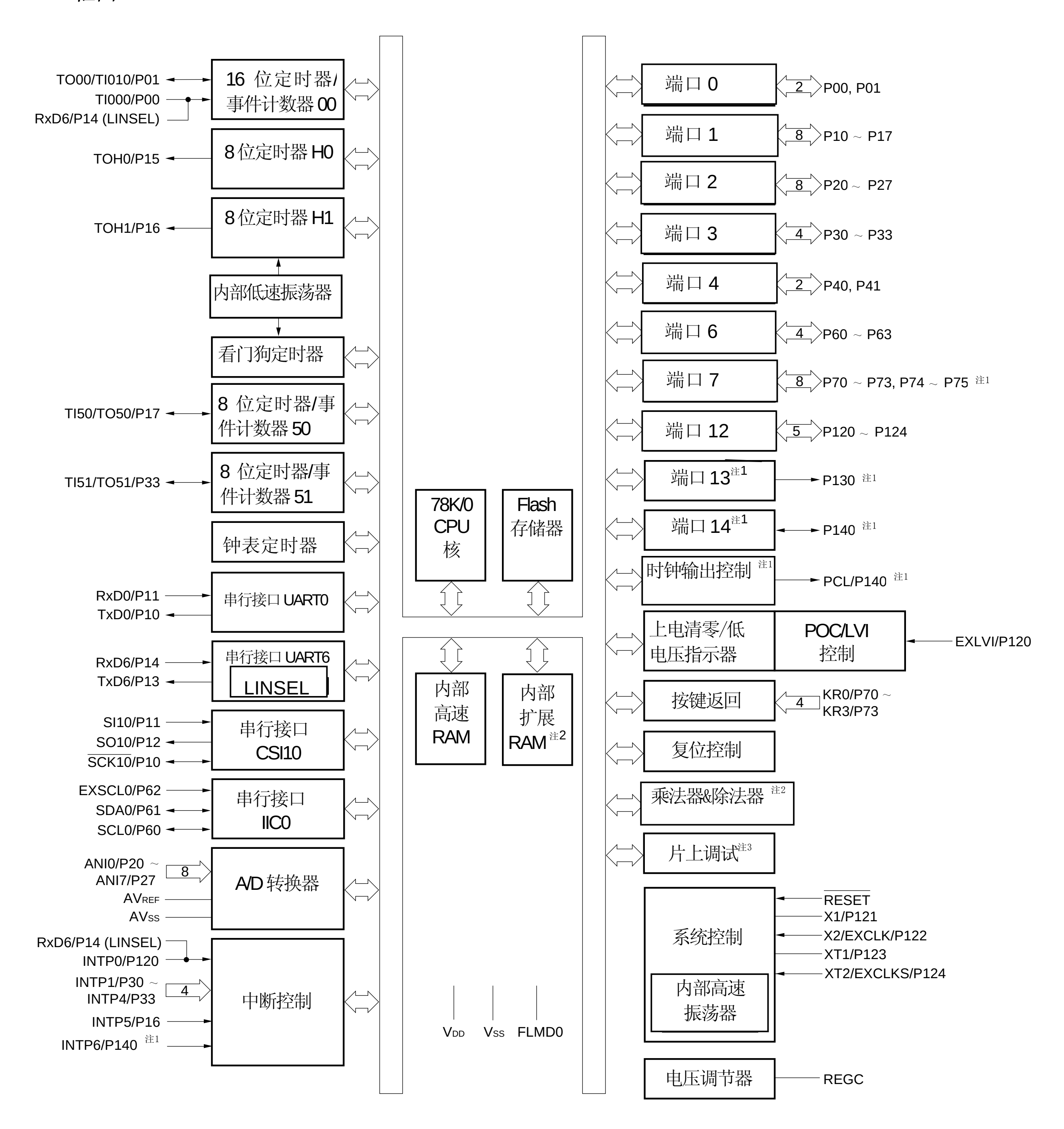
**注**　选择这些复用功能引脚中的任意一种功能。

(2/2)

| 项目 / 型号 | | 78K0/KD2 | | | | | | | 78K0/KE2 | | | | | | | 78K0/KF2 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 52 引脚 | | | | | | | 64 引脚 | | | | | | | 80 引脚 | | | |
| Flash 存储器 (KB) | | 16 | 24 | 32 | 48 | 60 | 96 | 128 | 16 | 24 | 32 | 48 | 60 | 96 | 128 | 48 | 60 | 96 | 128 |
| RAM (KB) | | 0.75 | 1 | 1 | 2 | 3 | 5 | 7 | 0.75 | 1 | 1 | 2 | 3 | 5 | 7 | 2 | 3 | 5 | 7 |
| Bank (flash 存储器) | | – | | | | | 4 | 6 | – | | | | | 4 | 6 | – | | 4 | 6 |
| 电源电压 | | $V_{DD}$ = 1.8 ～ 5.5 V | | | | | | | | | | | | | | | | | |
| 稳压器 | | 提供 | | | | | | | | | | | | | | | | | |
| 最小指令执行时间 | | 0.1 μs (20 MHz: $V_{DD}$ = 4.0 ～ 5.5 V)/0.2 μs (10 MHz: $V_{DD}$ = 2.7 ～ 5.5 V)/ 0.4 μs (5 MHz: $V_{DD}$ = 1.8 ～ 5.5 V) | | | | | | | | | | | | | | | | | |
| 时钟 | 主系统时钟 高速系统 | 20 MHz: $V_{DD}$ = 4.0 ～ 5.5 V/10 MHz: $V_{DD}$ = 2.7 ～ 5.5 V/5 MHz: $V_{DD}$ = 1.8 ～ 5.5 V | | | | | | | | | | | | | | | | | |
| 时钟 | 主系统时钟 内部高速振荡器 | 8 MHz (TYP.): $V_{DD}$ = 1.8 ～ 5.5 V | | | | | | | | | | | | | | | | | |
| 时钟 | 副时钟 | 32.768 kHz (TYP.): $V_{DD}$ = 1.8 ～ 5.5 V | | | | | | | | | | | | | | | | | |
| 时钟 | 内部低速振荡器 | 240 kHz (TYP.): $V_{DD}$ = 1.8 ～ 5.5 V | | | | | | | | | | | | | | | | | |
| 端口 | 合计 | 45 | | | | | | | 55 | | | | | | | 71 | | | |
| 端口 | N-ch O.D. (6 V 耐压) | 4 | | | | | | | 4 | | | | | | | 4 | | | |
| 定时器 | 16 位(TM0) | 1 通道 | | | | | | | | | | 2 通道 | | | | | | | |
| 定时器 | 8 位(TM5) | 2 通道 | | | | | | | | | | | | | | | | | |
| 定时器 | 8 位(TMH) | 2 通道 | | | | | | | | | | | | | | | | | |
| 定时器 | 钟表 | 1 通道 | | | | | | | | | | | | | | | | | |
| 定时器 | WDT 看门狗 | 1 通道 | | | | | | | | | | | | | | | | | |
| 串行接口 | 3 线 CSI | – | | | | | | | | | | 1 通道 | | | | | | | |
| 串行接口 | 自动收发的 3 线 CSI | – | | | | | | | | | | | | | | 1 通道 | | | |
| 串行接口 | UART/3 线CSI注 | 1 通道 | | | | | | | | | | | | | | | | | |
| 串行接口 | 支持 LIN 总线的 UART | 1 通道 | | | | | | | | | | | | | | | | | |
| 串行接口 | $I^2C$总线 | 1 通道 | | | | | | | | | | | | | | | | | |
| 10 位 A/D | | 8 通道 | | | | | | | | | | | | | | | | | |
| 中断 | 外部 | 8 | | | | | | | 9 | | | | | | | | | | |
| 中断 | 内部 | 16 | | | | | | | | | | 19 | | | | 20 | | | |
| 按键中断 | | 8 通道 | | | | | | | | | | | | | | | | | |
| 复位 | $\overline{RESET}$ 引脚 | 提供 | | | | | | | | | | | | | | | | | |
| 复位 | POC （上电清零） | 1.59 V ±0.15 V (电压上升到 1.8 V 所用时间: 3.6 ms (最小)) | | | | | | | | | | | | | | | | | |
| 复位 | LVI （低电压检测） | 可选择 16 个电源电压检测级别 | | | | | | | | | | | | | | | | | |
| 复位 | WDT（看门狗定时器） | 提供 | | | | | | | | | | | | | | | | | |
| 时钟输出/蜂鸣器输出 | | 仅有时钟输出 | | | | | | | 提供 | | | | | | | | | | |
| 乘法器/除法器 | | – | | | 提供 | | | | – | | | 提供 | | | | | | | |
| 片上调试功能 | | 仅μPD78F0527D | | | | | | | 仅μPD78F0537D | | | | | | | 仅μPD78F0547D | | | |
| 工作环境温度 | | $T_A$ = –40 ～ +85°C | | | | | | | | | | | | | | | | | |

**注**　选择这些复用功能引脚中的任意一种功能。

## 1.6 框图




注　**1.** 仅在 48 引脚产品可用。
**2.** 仅在 μPD78F0514, 78F0515, 和 78F0515D 中可用。
**3.** 仅在 μPD78F0513D 和 78F0515D 中可用。

## 1.7 功能概述

(1/2)

| 项目 | | μPD78F0511 | μPD78F0512 | μPD78F0513 | μPD78F0513D | μPD78F0514 | μPD78F0515 | μPD78F0515D |
|---|---|---|---|---|---|---|---|---|
| 内部存储器(字节) | Flash 存储器 (支持自编程)[注] | 16 K | 24 K | 32 K | | 48 K | 60 K | |
| | 高速 RAM[注] | 768 | 1 K | | | | | |
| | 扩展 RAM[注] | – | | | | 1 K | 2 K | |
| 存储空间 | | 64 KB | | | | | | |
| 主系统时钟 (振荡频率) | 高速系统时钟 | X1 (晶体/陶瓷) 振荡器， 外部主系统时钟输入( EXCLK )<br>1 ～ 20 MHz: $V_{DD}$ = 4.0 ～ 5.5 V, 1 ～ 10 MHz: $V_{DD}$ = 2.7 ～ 5.5 V,<br>1 ～ 5 MHz: $V_{DD}$ = 1.8 ～ 5.5 V | | | | | | |
| | 内部高速振荡时钟 | 内部振荡器<br>8 MHz (TYP.): $V_{DD}$ = 1.8 ～ 5.5 V | | | | | | |
| 副时钟 (振荡频率) | | XT1 (晶体) 振荡器，外部副时钟输入( EXCLKS )<br>32.768 kHz (TYP.): $V_{DD}$ = 1.8 ～ 5.5 V | | | | | | |
| 内部低速振荡时钟(用于 TMH1, WDT) | | 内部振荡器<br>240 kHz (TYP.): $V_{DD}$ = 1.8 ～ 5.5 V | | | | | | |
| 通用寄存器 | | 8 位 × 32 个寄存器 (8 位 × 8 个寄存器 × 4 banks) | | | | | | |
| 最小指令执行时间 | | 0.1 μs/0.2 μs/0.4 μs/0.8 μs/1.6 μs (高速系统时钟: @ $f_{XH}$ = 20 MHz 工作) | | | | | | |
| | | 0.25 μs/0.5 μs/1.0 μs/2.0 μs/4.0 μs (TYP.) (内部高速振荡器: @ $f_{RH}$ = 8 MHz (TYP.) 工作) | | | | | | |
| | | 122 μs (副时钟: @ $f_{SUB}$ = 32.768 kHz 工作) | | | | | | |
| 指令集 | | • 16 位操作<br>• 乘/除 (8 位 × 8 位, 16 位 ÷ 8 位)<br>• 位操作 (置位，复位，测试和布尔操作)<br>• BCD 调整，等 | | | | | | |
| I/O 端口 | | 合计: 37 (44 引脚产品) 41(48 引脚产品)<br>CMOS I/O: 32 36<br>CMOS 输出: 1 1<br>N沟开漏 I/O (6 V 耐压) : 4 4 | | | | | | |
| 定时器 | | • 16 位定时器/时间计数器: 1 通道<br>• 8 位定时器/时间计数器: 2 通道<br>• 8 位定时器: 2 通道<br>• 钟表定时器: 1 通道<br>• 看门狗定时器: 1 通道 | | | | | | |
| | 定时器输出 | 5 (PWM 输出: 4) | | | | | | |
| 时钟输出 (仅 48 引脚产品) | | • 156.25 kHz, 312.5 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz<br>(外围设备硬件时钟: @ $f_{PRS}$ = 20 MHz 工作)<br>• 32.768 kHz (副时钟: @ $f_{SUB}$ = 32.768 kHz 工作) | | | | | | |
| A/D 转换器 | | 10 位分辨率 × 8 通道 ($AV_{REF}$ = 2.3 ～ 5.5 V) | | | | | | |

注　通过使用内部存储器容量切换寄存器(IMS)和内部扩展 RAM 容量切换寄存器(IXS)，可以改变内部 Flash 存储器容量、内部高速 RAM 容量和内部扩展 RAM 的容量。

(2/2)

| 项目 | | μPD78F0511 | μPD78F0512 | μPD78F0513 | μPD78F0513D | μPD78F0514 | μPD78F0515 | μPD78F0515D |
|---|---|---|---|---|---|---|---|---|
| 串行接口 | | • 支持LIN总线的 UART模式: 1 通道<br>• 3 线串行 I/O 模式/UART 模式[注]: 1 通道<br>• $I^2C$ 总线模式: 1 通道 | | | | | | |
| 乘法器/除法器 | | – | | | | • 16 位 × 16 位 = 32 位 (乘法)<br>• 32 位 ÷ 16 位 = 32 位 余数 16 位 (除法) | | |
| 矢量中断源 | 内部 | 16 | | | | | | |
| | 外部 | 7 (44 引脚产品), 8 (引脚产品) | | | | | | |
| 按键中断 | | 通过检测按键输入引脚(KR0 ～ KR3)的下降沿产生按键中断 (INTKR) | | | | | | |
| 复位 | | • 使用 $\overline{\text{RESET}}$ 引脚复位<br>• 使用看门狗定时器内部复位<br>• 使用上电清零电路内部复位<br>• 使用低电压检测电路内部复位 | | | | | | |
| 片上调试功能 | | – | | | 提供 | – | | 提供 |
| 电源电压 | | $V_{DD}$ = 1.8 ～ 5.5 V | | | | | | |
| 工作环境温度 | | $T_A$ = –40 ～ +85°C | | | | | | |
| 封装形式 | | • 44 引脚塑封 LQFP (10 × 10)<br>• 48 引脚塑封 LQFP (密脚距) (7 × 7) | | | | | | |

注　选择这些复用功能引脚中的任意一种功能。

定时器概要说明

| | | 16 位定时器/事件计数器 00 | 8 位定时器/事件计数器 50 和 51 | | 8 位定时器 H0 和 H1 | | 钟表定时器 | 看门狗定时器 |
|---|---|---|---|---|---|---|---|---|
| | | TM00 | TM50 | TM51 | TMH0 | TMH1 | | |
| 功能 | 间隔定时器 | 1 通道 | 1 通道 | 1 通道 | 1 通道 | 1 通道 | 1 通道[注1] | – |
| | 外部事件计数器 | 1 通道 | 1 通道 | 1 通道 | – | – | – | – |
| | PPG 输出 | 1 输出 | – | – | – | – | – | – |
| | PWM 输出 | – | 1 输出 | 1 输出 | 1 输出 | 1 输出 | – | – |
| | 脉冲宽度测量 | 2 输入 | – | – | – | – | – | – |
| | 方波输出 | 1 输出 | 1 输出 | 1 输出 | 1 输出 | 1 输出 | – | – |
| | 载波发生器 | – | – | – | – | 1 输出[注2] | – | – |
| | 钟表定时器 | – | – | – | – | – | 1 通道[注1] | – |
| | 看门狗定时器 | – | – | – | – | – | – | 错误！链接无效。 |
| 中断源 | | 2 | 1 | 1 | 1 | 1 | 1 | – |

注　**1.**　在钟表定时器中，看门狗定时器功能和间隔定时器功能同时可用。

　　**2.**　TM51 和 TMH1 可联合使用，作为载波发生器模式。

# 第二章　引脚功能

## 2.1 引脚功能列表

有两种类型的引脚 I/O 缓冲器电源: $AV_{REF}$ 和 $V_{DD}$. 下表显示了这些供电电源与引脚之间的关系。

**表 2-1. 引脚 I/O 缓冲器供电电源**

| 电源电压 | 对应引脚 |
| --- | --- |
| $AV_{REF}$ | P20 ～ P27 |
| $V_{DD}$ | 除去 P20 ～ P27 的其它引脚 |

**(1) 端口功能 (1/2)**

| 功能名称 | I/O | 功能 | 复位后 | 复用功能 |
| --- | --- | --- | --- | --- |
| P00 | I/O | 端口 0<br>2 位 I/O 端口<br>可以位选输入/输出模式<br>通过软件设置，可以定义内置上拉电阻的使用 | 输入端口 | TI000 |
| P01 | | | | TI010/TO00 |
| P10 | I/O | 端口 1<br>8 位 I/O 端口<br>可以位选输入/输出模式<br>通过软件设置，可以定义内置上拉电阻的使用 | 输入端口 | $\overline{\text{SCK10}}$/TxD0 |
| P11 | | | | SI10/RxD0 |
| P12 | | | | SO10 |
| P13 | | | | TxD6 |
| P14 | | | | RxD6 |
| P15 | | | | TOH0 |
| P16 | | | | TOH1/INTP5 |
| P17 | | | | TI50/TO50 |
| P20 ～ P27 | I/O | 端口 2<br>8 位 I/O 端口<br>可以位选输入/输出模式 | 模拟输入 | ANI0 ～ ANI7 |
| P30 | I/O | 端口 3<br>4 位 I/O 端口<br>可以位选输入/输出模式<br>通过软件设置，可以定义内置上拉电阻的使用 | 输入端口 | INTP1 |
| P31 | | | | INTP2/OCD1A[注] |
| P32 | | | | INTP3/OCD1B[注] |
| P33 | | | | TI51/TO51/INTP4 |

注　仅限μPD78F0513D 和 78F0515D

(1) 端口功能 (2/2)

| 功能名称 | I/O | 功能 | 复位后 | 复用功能 |
|---|---|---|---|---|
| P40 和 P41 | I/O | 端口 4<br>2 位 I/O 端口<br>可以位选输入/输出模式<br>通过软件设置，可以定义内置上拉电阻的使用 | 输入端口 | – |
| P60 | I/O | 端口 6<br>4 位 I/O 端口<br>P60 ～ P63 输出是 N 沟开漏输出(6V 耐压)<br>可以位选输入/输出模式 | 输入端口 | SCL0 |
| P61 | | | | SDA0 |
| P62 | | | | EXSCL0 |
| P63 | | | | – |
| P70 ～ P73 | I/O | 端口 7<br>6 位 I/O 端口<br>可以位选输入/输出模式<br>通过软件设置，可以定义内置上拉电阻的使用 | 输入 | KR0 ～ KR3 |
| P74[注 1] 和P75[注 1] | | | | – |
| P120 | I/O | 端口 12<br>5 位 I/O 端口<br>可以位选输入/输出模式<br>只有 P120 引脚可以通过软件设置，可以定义内置上拉电阻的使用 | 输入端口 | INTP0/EXLVI |
| P121 | | | | X1/OCD0A[注 2] |
| P122 | | | | X2/EXCLK/OCD0B[注 2] |
| P123 | | | | XT1 |
| P124 | | | | XT2/EXCLKS |
| P130[注 1] | 输出 | 端口 13<br>1 位 输出端口 | 输出端口 | – |
| P140[注 1] | I/O | 端口 14<br>1 位 I/O 端口<br>可以位选输入/输出模式<br>通过软件设置，可以定义内置上拉电阻的使用 | 输入端口 | PCL/INTP6[注 1] |

注 **1.** 仅限 48 引脚产品
**2.** 仅限$\mu$PD78F0513D 和 78F0515D

**(2) 非端口功能 (1/2)**

| 功能名称 | I/O | 功能 | 复位后 | 复用功能 |
|---|---|---|---|---|
| ANI0 ～ ANI7 | 输入 | A/D 转换器模拟输入 | 模拟输入 | P20 ～ P27 |
| EXLVI | 输入 | 用于外部低电压检测的电压输入 | 输入端口 | P120/INTP0 |
| EXSCL0 | 输入 | 用于串行接口的外部时钟输入<br>为输入外部时钟，输入一个 6.4 MHz的时钟。 | 输入端口 | P62 |
| FLMD0 | – | Flash 存储器编程模式设置 | – | – |
| INTP0 | 输入 | 定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），用于外部中断请求输入 | 输入端口 | P120/EXLVI |
| INTP1 | | | | P30 |
| INTP2 | | | | P31/OCD1A[注1] |
| INTP3 | | | | P32/OCD1B[注1] |
| INTP4 | | | | P33/TI51/TO51 |
| INTP5 | | | | P16/TOH1 |
| INTP6[注2] | | | | P140/PCL[注2] |
| KR0 ～ KR3 | 输入 | 按键中断输入 | 输入端口 | P70 ～ P73 |
| PCL[注2] | 输出 | 时钟输出（用于高速系统时钟、副时钟的调整） | 输入端口 | P140/INTP6[注2] |
| REGC | – | 将稳压器输出 (2.5V) 连接至稳定电容，用于内部操作。<br>通过一个电容器 (0.47 μF ～ 1μF: 推荐) 连接至 $V_{SS}$。. | – | – |
| $\overline{\text{RESET}}$ | 输入 | 系统复位输入 | – | – |
| RxD0 | 输入 | 串行数据输入到异步串行接口 | 输入端口 | P11/SI10 |
| RxD6 | | | | P14 |
| $\overline{\text{SCK10}}$ | I/O | 用于串行接口的时钟输入/输出 | 输入端口 | P10/TxD0 |
| SCL0 | | | | P60 |
| SDA0 | I/O | 用于串行接口的串行数据 I/O | 输入端口 | P61 |
| SI10 | 输入 | 串行接口的串行数据输入 | 输入端口 | P11/RxD0 |
| SO10 | 输出 | 串行接口的串行数据输出 | 输入端口 | P12 |
| TI000 | 输入 | 16 位定时器/事件计数器 00 的外部计数时钟输入<br>16 位定时器/事件计数器 00 的捕捉寄存器(CR000、CR010)的捕捉触发输入 | 输入端口 | P00 |
| TI010 | | 16 位定时器/事件计数器 00 的捕捉寄存器(CR000)的捕捉触发输入 | | P01/TO00 |
| TI50 | 输入 | 8 位定时器/事件计数器 50 的外部计数时钟输入 | 输入端口 | P17/TO50 |
| TI51 | | 8 位定时器/事件计数器 51 的外部计数时钟输入 | | P33/TO51/INTP4 |
| TO00 | 输出 | 16 位定时器/事件计数器 00 输出 | 输入端口 | P01/TI010 |
| TO50 | 输出 | 8 位定时器/事件计数器 50 输出 | 输入端口 | P17/TI50 |
| TO51 | | 8 位定时器/事件计数器 51 输出 | | P33/TI51/INTP4 |
| TOH0 | 输出 | 8 位定时器 H0 输出 | 输入端口 | P15 |
| TOH1 | | 8 位定时器 H1 输出 | | P16/INTP5 |

注　**1.** 仅限μPD78F0513D 和 78F0515D
　　**2.** 仅限 48 引脚产品

**(2) 非端口功能 (2/2)**

| 功能名称 | I/O | 功能 | 复位后 | 复用功能 |
|---|---|---|---|---|
| TxD0 | 输出 | 异步串行接口的串行数据输出 | 输入端口 | P10/$\overline{SCK10}$ |
| TxD6 | | | | P13 |
| X1 | 输入 | 连接主系统时钟振荡器 | 输入端口 | P121/OCD0A[注] |
| X2 | – | | | P122/EXCLK/OCD0B[注] |
| EXCLK | 输入 | 主系统时钟的外部时钟输入 | 输入端口 | P122/X2/OCD0B[注] |
| XT1 | 输入 | 连接副时钟振荡器 | 输入端口 | P123 |
| XT2 | – | | 输入端口 | P124/EXCLKS |
| EXCLKS | 输入 | 副时钟的外部时钟输入 | 输入端口 | P124/XT2 |
| $V_{DD}$ | – | 驱动引脚的正向电源，P20 ～ P27 除外 | – | – |
| $AV_{REF}$ | 输入 | A/D 转换器的参考电压输入和 P20 ～ P27 及 A/D 转换器的正向供电电源 | – | – |
| $V_{SS}$ | – | 端口地电位，P20 ～ P27 除外 | – | – |
| $AV_{SS}$ | – | A/D 转换器的地电位。应与 $V_{SS}$ 的电位相同 | – | – |
| OCD0A[注] | 输入 | 用于设置片上调试模式 | 输入端口 | P121/X1 |
| OCD1A[注] | | （仅用于 PD78F0513D 和 78F0515D） | | P31/INTP2 |
| OCD0B[注] | – | | | P122/X2/EXCLK |
| OCD1B[注] | | | | P32/INTP3 |

注　　仅限μPD78F0513D 和 78F0515D

## 2.2 引脚功能描述

### 2.2.1 P00 和 P01 (端口 0)

P00 和 P01 作为 2 位 I/O 端口使用。这些引脚也可用于定时器 I/O。
以下操作模式可以用位选指定。

**(1) 端口模式**

P00 和 P01 作为 2 位 I/O 端口使用。通过使用端口模式寄存器 0 (PM0)，可按位设置 P00 和 P01 为输入或输出端口。由上拉电阻选择寄存器 0 (PU0)规定内置上拉电阻的使用。

**(2) 控制模式**

P00 和 P01 作为定时器 I/O 的功能。

**(a) TI000**

将外部计数时钟输入到 16 位定时器/事件计数器 00，也可将捕捉触发信号输入到 16 位定时器/事件计数器 00 的捕捉寄存器（CR000、 CR010）中。

**(b) TI010**

将捕捉触发信号输入到 16 位定时器/事件计数器 00 的捕捉寄存器（CR000）中。

**(c) TO00**

用于 16 位定时器/时间计数器 00 的定时器输出。

### 2.2.2 P10 到 P17 (端口 1)

P10 至 P17 作为 8 位 I/O 端口使用。这些引脚也可用于外部中断请求输入、串行接口数据 I/O、时钟 I/O 以及定时器 I/O。

以下操作模式可以用位选指定。

**(1) 端口模式**

P10 至 P17 作为 8 位 I/O 端口使用。通过使用端口模式寄存器 1 (PM1)，可按位设置 P10 至 P17 为输入输出端口。 由上拉电阻选择寄存器 1 (PU1)定义内置上拉电阻的使用。

**(2) 控制模式**

P10 至 P17 可用于外部中断请求输入、串行接口数据 I/O、时钟 I/O 和定时器 I/O。

**(a) SI10**

用于串行接口 CSI10 的串行数据输入。

**(b) SO10**

用于串行接口 CSI10 的串行数据输出。

**(c) $\overline{\text{SCK10}}$**

用于串行接口 CSI10 的串行时钟 I/O 。

**(d) RxD0**

用于串行接口 UART0 的串行数据输入。

**(e) RxD6**

用于串行接口 UART6 的串行数据输入。

**(f) TxD0**

用于串行接口 UART0 的串行数据输出。

**(g) TxD6**

用于串行接口 UART6 的串行数据输出。

**(h) TI50**

用于输入一个外部计数时钟到 8 位定时器/事件计数器 50。

**(i) TO50**

用于 8 位定时器/事件计数器 50 的定时器输出。

**(j) TOH0, TOH1**

用于 8 位定时器 H0 和 H1 的定时器输出。

**(k) INTP5**

可定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），用于外部中断请求输入。

### 2.2.3 P20 到 P27 (端口 2)

P20 至 P27 作为 8 位 I/O 端口使用，也可用于 A/D 转换器模拟输入。

以下操作模式可以用位选指定。

**(1)** 端口模式

P20 至 P27 作为 8 位 I/O 端口使用。通过使用端口模式寄存器 2 (PM2)，可按位设置 P20 至 P27 为输入输出端口。

**(2)** 控制模式

P20 至 P27 用于 A/D 转换器模拟输入引脚（ANI0 至 ANI7），可参见 **12.6 A/D 转换器注意事项**中**(5) ANI0/P20** 至 **ANI7/P27** 。

注意事项　**复位后，ANI0/P20** 至 **ANI7/P27 被设置为模拟输入模式。**

### 2.2.4 P30 到 P33 (端口 3)

P30 至 P33 作为 4 位 I/O 端口使用，也可用于外部中断请求输入和定时器 I/O。

以下操作模式可以用位选指定。

**(1)**端口模式

P30 至 P33 作为 4 位 I/O 端口使用。 通过使用端口模式寄存器 3 (PM3)，可按位设置 P30 至 P33 为输入输出端口。由上拉电阻选择寄存器 3 (PU3)定义内置上拉电阻的使用。

**(2)** 控制模式

P30 至 P33 用于外部中断请求输入和定时器 I/O。

**(a) INTP1** 至 **INTP4**

可定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），用于外部中断请求输入。

**(b) TI51**

将外部计数时钟输入到 8 位定时器/事件计数器 51 中。

**(c) TO51**

用于 8 位定时器/事件计数器 51 的定时器输出。

注意事项 **1. 在使用具有片上调试功能的产品 (μPD78F0513D 和 78F0515D) 时，应确保在复位以前将引脚 P31/INTP2/OCD1A**[注] **下拉，防止故障发生。**

<R> 注意事项 **2.** 对于具有 **48KB**或更大**Flash**存储器而没有片上调试功能的产品 **(μPD78F0514 和 78F0515)**，有一个**"I", "K", 或 "E"** 的产品等级，对于具有片上调试功能的产品 **(μPD78F0513D 和 78F0515D)** 在使用**Flash**编程器写**Flash**存储器时，应按以下连接**P31/INTP2/OCD1A**[注]。

- **P31/INTP2/OCD1A**[注]**:** 通过电阻连接到 **Vss (10 kΩ: 推荐)。**

当通过自编程方式写入 **Flash** 存储器时，以上连接并不需要。

<R> 注　只有μPD78F0513D 和 78F0515D 提供 OCD1A 。

<R> 备注 **1.** 对于产品级别，请咨询 NEC 电子的销售代表。

**2.** 只有μPD78F0513D 和 78F0515D，在使用片上调试功能时，P31 和 P32 能够用于片上调试模式设定引脚 (OCD1A, OCD1B) 。对于怎样连接支持片上调试功能的在线仿真器 (QB-78K0MINI)，请参看 **第 26 章 片上调试功能 (仅μPD78F0513D 和 78F0515D)** 。

### 2.2.5 P40 到 P41 (端口 4)

P40 和 P41 作为 2 位 I/O 端口使用。通过使用端口模式寄存器 4 (PM4)，可按位设置 P40 和 P41 为输入输出端口。由上拉电阻选择寄存器 4 (PU4)定义内置上拉电阻的使用。

### 2.2.6 P60 到 P63 (端口 6)

P60 至 P63 作为 4 位 I/O 端口使用，也可作为串行接口数据 I/O、时钟 I/O 和外部时钟输入引脚。

以下操作模式可以用位操作。

**(1)** 端口模式

P60 至 P63 作为 4 位 I/O 端口使用。通过使用端口模式寄存器 6 (PM6)，可按位设置 P60 至 P63 为输入输出端口。

P60 至 P63 输出为 N-ch 漏极开路输出（6 V 耐压）。

**(2)** 控制模式

P60 至 P63 作为串行接口数据 I/O，时钟 I/O 和外部时钟输入使用。

**(a) SDA0**

用于串行接口 IIC0 的串行数据 I/O。

**(b) SCL0**

用于串行接口 IIC0 的串行时钟 I/O。

**(c) EXSCL0**

将外部时钟输入到串行接口 IIC0 中。为输入外部时钟，需输入一个频率为 6.4MHz 的时钟。

### 2.2.7 P70 到 P75 (端口 7)

P70 至 P75 作为 6 位 I/O 端口使用，P70 至 P73 也可用于按键中断输入。

以下操作模式可以用位操作

**(1)** 端口模式

P70 至 P75 作为 6 位 I/O 端口使用。通过使用端口模式寄存器 7(PM7)，可按位设置 P70 至 P77 为输入输出端口。由上拉电阻选择寄存器 7(PU7)定义内置上拉电阻的使用。

**(2)** 控制模式 **(**仅 **P70** 至 **P73)**

P70 至 P73 用于按键中断输入。

**(a) KR0** 至 **KR3**

用于按键中断输入。

备注 在 44 引脚的产品中提供 P70 至 P73 引脚，在在 48 引脚的产品中提供 P70 至 P75 引脚。

### 2.2.8 P120 到 P124 (端口 12)

P120 至 P124 作为 5 位 I/O 端口使用。也可以作为外部中断请求输入、外部低电压检测的电压输入、连接主系统时钟振荡器、连接副时钟振荡器、主系统时钟的外部时钟输入和副时钟的外部时钟输入。以下操作模式可以用位操作。

**(1)** 端口模式

P120 至 P124 作为 5 位 I/O 端口使用，可由端口模式寄存器 12(PM12)，将 P120 至 P124 设置为输入或输出端口。对于 P120，可由上拉电阻选择寄存器 12(PU12)定义内置上拉电阻的使用。

**(2)** 控制模式

P120 至 P124 可作为外部中断请求输入、外部低电压检测的电压输入、连接主系统时钟的振荡器、连接副时钟的振荡器、主系统时钟的外部时钟输入和副时钟的外部时钟输入使用。

**(a) INTP0**

通过定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），它可作为外部中断请求输入(INTP0)使用。

**(b) EXLVI**

用于外部低电压检测的电压输入。

**(c) X1, X2**

用于连接主系统时钟振荡器。

**(d) EXCLK**

用于主系统时钟的外部时钟输入。

**(e) XT1, XT2**

用于连接副时钟振荡器。

**(f) EXCLKS**

用于副时钟的外部时钟输入。

<R> 注意事项　对于具有 **48KB**或更大**Flash**存储器而没有片上调试功能的产品 **(μPD78F0514** 和 **78F0515)**，有一个**“I”，“K”，**或 **“E”** 的产品等级，对于具有片上调试功能的产品 **(μPD78F0513D** 和 **78F0515D)** 在使用**Flash**编程器写**Flash**存储器时，应按以下连接**P121/X1/OCD0A**[注]。

- **P121/X1/OCD0A**[注]**:** 当使用该引脚作为一个端口时，通过电阻连接到 **V**SS **(10 kΩ:** 推荐**)(**在输入模式**)**或悬空**(**在输出模式**)**。

当通过自编程方式写入 **Flash** 存储器时，以上连接并不需要。

<R> 注　只有μPD78F0513D 和 78F0515D 提供 OCD0A 。

<R> 备注　**1.** 对于产品级别，请咨询 NEC 电子的销售代表。

**2.** 只有μPD78F0513D 和 78F0515D，在使用片上调试功能时，X1 和 X2 能够用于片上调试模式设定引脚 (OCD0A, OCD0B) 。对于怎样连接支持片上调试功能的在线仿真器 (QB-78K0MINI) ，请参看 **第 26 章　片上调试功能 (**仅限**μPD78F0513D** 和 **78F0515D)** 。

### 2.2.9 P130 (端口 13) (仅 48 引脚产品)

P130 作为 1 位输出端口使用。

备注　当设备复位后，P130 输出一个低电平。因此，在设备复位前使 P130 输出高电平，P130 的输出信号可用作虚拟的 CPU 复位信号(参见 **4.2.9　端口 13 备注中**的图(仅限 48 引脚产品))。

### 2.2.10 P140 (端口 14) (仅 48 引脚产品)

P140 作为 1 位 I/O 端口使用。也可用于外部中断请求输入、时钟输出。
以下操作模式可以用位操作。

**(1)** 端口模式

P140 作为 1 位 I/O 端口使用。通过使用端口模式寄存器 14(PM14)，可按位设置 P140 为输入输出端口。由上拉电阻选项寄存器 14(PU14)，可定义内置上拉电阻的使用。

**(2)** 控制模式

P140 用于外部中断请求输入和时钟输出。

**(a) INTP6**

通过定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），可用于外部中断请求输入。

**(b) PCL**

用于时钟输出。

### 2.2.11 $AV_{REF}$

用于 A/D 转换器参考电压输入和 P20 至 P27 及 A/D 转换器的正向电源供电。

不使用A/D转换器时，将该引脚直接连到$V_{DD}$[注]。

**注** 当端口 2 作为数字端口使用时，应使 $AV_{REF}$ 引脚电平与 $V_{DD}$ 引脚的电平相同。

### 2.2.12 $AV_{SS}$

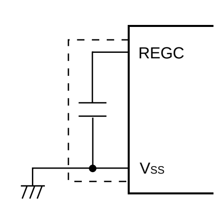
作为 A/D 转换器的地引脚。即使在不使用 A/D 转换器时，该引脚电平也始终应与 $V_{SS}$ 相同。

### 2.2.13 $\overline{RESET}$

用于系统复位输入，低电平有效。

### 2.2.14 REGC

<R> 用于内部操作的调节器输出(2.5 V)稳定电容的连接。通过一个电容 (0.47 ～1 μF : 推荐) 将此引脚连接到 $V_{SS}$ 。




**注意事项　在上图虚线部分，用尽可能短的配线连接。**

### 2.2.15 $V_{DD}$

$V_{DD}$ 为正向供电电源引脚（P20 ～ P27 端口除外）。

### 2.2.16 $V_{SS}$

$V_{SS}$ 为地引脚（P20 ～ P27 端口除外）。

### 2.2.17 FLMD0

用于 Flash 编程模式设置。

在正常操作模式下将 FLMD0 连接至 $V_{SS}$ 。

在 Flash 编程模式下，将该引脚连接至 Flash 编程器。

## 2.3 引脚 I/O 电路和未使用引脚的建议连接方式

表 2-2 为引脚 I/O 电路类型和未使用引脚的建议连接方式。

参考 **图 2-1** 显示每一种类型的 I/O 电路结构。

**表 2-2. 引脚 I/O 电路类型 (1/2)**

| 引脚名称 | I/O 电路类型 | I/O | 未使用引脚的建议连接方式 |
|---|---|---|---|
| P00/TI000 | 5-AH | I/O | 输入：　通过电阻单独连接至 $V_{DD}$ 或 $V_{SS}$ 。<br>输出：　保持开路。 |
| P01/TI010/TO00 | | | |
| P10/$\overline{SCK10}$/TxD0 | | | |
| P11/SI10/RxD0 | | | |
| P12/SO10 | 5-AG | | |
| P13/TxD6 | | | |
| P14/RxD6 | 5-AH | | |
| P15/TOH0 | 5-AG | | |
| P16/TOH1/INTP5 | 5-AH | | |
| P17/TI50/TO50 | | | |
| ANI0/ P20 ～ ANI7/P27[注1] | 11-G | | <模拟设置><br>连接至 $AV_{REF}$ 或 $AV_{SS}$ 。<br><数字设置><br>输入：　通过电阻单独连接至 $V_{DD}$ 或 $V_{SS}$ 。<br>输出：　保持开路。 |
| P30/INTP1 | 5-AH | | 输入：　通过电阻单独连接至 $V_{DD}$ 或 $V_{SS}$ 。<br>输出：　保持开路。 |
| P31/INTP2/OCD1A[注2, 3] | | | |
| P32/INTP3/OCD1B[注3] | | | |
| P33/TI51/TO51/INTP4 | | | |
| P40 and P41 | 5-AG | | |
| P60/SCL0 | 13-AD | | 输入：　连接至 $V_{SS}$ 。<br>输出：　将端口的输出锁存器清零后，该引脚保持低电平悬空输出 |
| P61/SDA0 | | | |
| P62/EXSCL0 | | | |
| P63 | 13-P | | |
| P70/KR0 ～ P73/KR3 | 5-AH | | 输入：　通过电阻单独连接至 $V_{DD}$ 或 $V_{SS}$ 。<br>输出：　保持开路。 |
| P74[注4] 和 P75[注4] | | | |
| P120/INTP0/EXLVI | | | |

注　**1.** 复位后，ANI0/P20 ～ ANI7/P27 被设定为模拟输入模式。

<R>　**2.** 对于具有 48KB或更大Flash存储器而没有片上调试功能的产品 ($\mu$PD78F0514 和 78F0515)，有一个“I”, “K”, 或 “E” 的产品等级，对于具有片上调试功能的产品 ($\mu$PD78F0513D 和 78F0515D) 在使用Flash编程器写Flash存储器时，应按以下连接P31/INTP2/OCD1A[注3]。

- P31/INTP2/OCD1A[注3]: 通过电阻连接到 $V_{SS}$ (10 kΩ: 推荐)。

当通过自编程方式写入 Flash 存储器时，以上连接并不需要。

<R>　**3.** 只有$\mu$PD78F0513D 和 78F0515D 提供 OCD1A 和 OCD1B 。

**4.** 只有 48 引脚产品提供 P74 和 P75 。

<R> **备注**　对于产品级别，请咨询 NEC 电子的销售代表。

**表 2-2. 引脚 I/O 电路类型 (2/2)**

| 引脚名称 | I/O 电路类型 | I/O | 未使用引脚的建议连接方式 |
|---|---|---|---|
| P121/X1[注 1,2,6] | 37 | I/O | 输入：　通过电阻单独连接至 $V_{DD}$ 或 $V_{SS}$ 。<br>输出：　保持开路。 |
| P122/X2/EXCLK[注 1,6] | | | |
| P123/XT1[注 1] | | | |
| P124/XT2/EXCLKS[注 1] | | | |
| P130[注 5] | 3-C | 输出 | 保持开路。 |
| P140/PCL/INTP6[注 5] | 5-AH | I/O | 输入：　通过电子单独连接至 $V_{DD}$ 或 $V_{SS}$ 。<br>输出：　保持开路。 |
| $AV_{REF}$ | – | – | 直连到 $V_{DD}$[注 3] 。 |
| $AV_{SS}$ | | | 直连到 $V_{SS}$ 。 |
| FLMD0 | 38 | 输入 | 连到$V_{SS}$[注 4] 。 |
| $\overline{RESET}$ | 2 | 输入 | 直接或通过一个电阻连到 $V_{DD}$ 。 |

<R>（RESET 行）

注　**1.** 当这些端口没有使用的情况下，在 I/O 端口模式下应按照以上的推荐连接 (参看 **图 5-2　时钟工作模式选择寄存器(OSCCTL)的格式**) 。

<R> **2.** 对于具有 48KB或更大Flash存储器而没有片上调试功能的产品 ($\mu$PD78F0514 和 78F0515) ，有一个“I”, “K”, 或 “E” 的产品等级，对于具有片上调试功能的产品 ($\mu$PD78F0513D 和 78F0515D) 在使用Flash编程器写Flash存储器时，应按以下连接P121/X1/OCD0A[注 5]。
- P121/X1/OCD0A[注 5]: 通过电阻连接到 $V_{SS}$ (10 kΩ: 推荐)(在输入模式)或者悬空(在输出模式)。

当通过自编程方式写入 Flash 存储器时，以上连接并不需要。

**3.** 当端口 2 被用做数字端口时，确保电平与 $V_{DD}$ 一致。

<R> **4.** FLMD0 是被用来向 Flash 存储器写数据的引脚。为了以 on-board 形式重写 flash 存储器的数据，应通过一个电阻 (10 kΩ: 推荐) 使此引脚与 $V_{SS}$ 相连。当使用带有片上调试功能的产品 ($\mu$PD78F0513D 和 78F0515D) 执行片上调试时，也要做此处理。

<R> **5.** 只有 48 引脚产品提供 P130 和 P140 。

<R> **6.** 只有$\mu$PD78F0513D 和 78F0515D 提供 OCD0A 和 OCD0B 。

<R> **备注**　对于产品级别，请咨询 NEC 电子的销售代表。

图 2-1. 引脚 I/O 电路列表 (1/2)

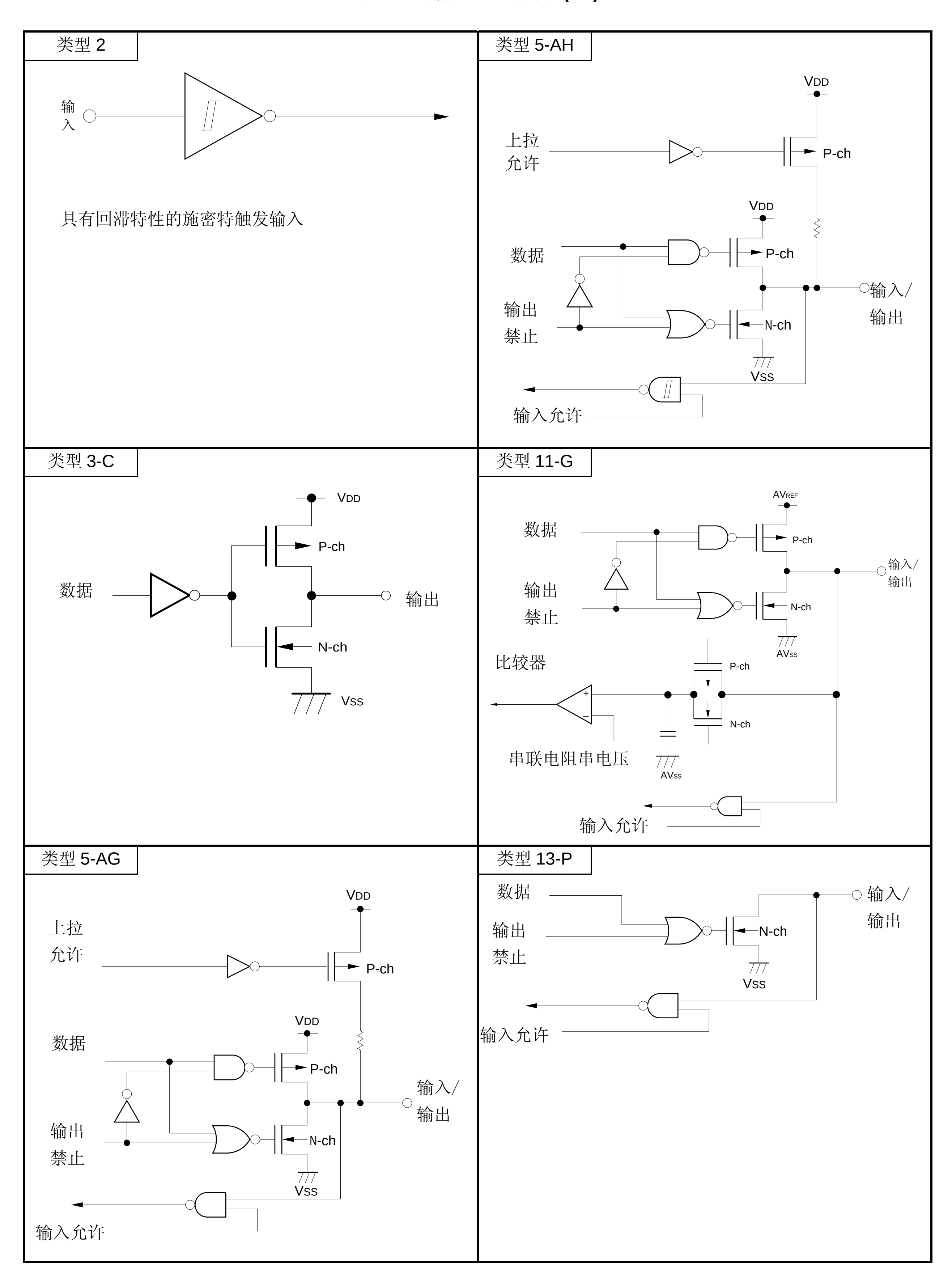
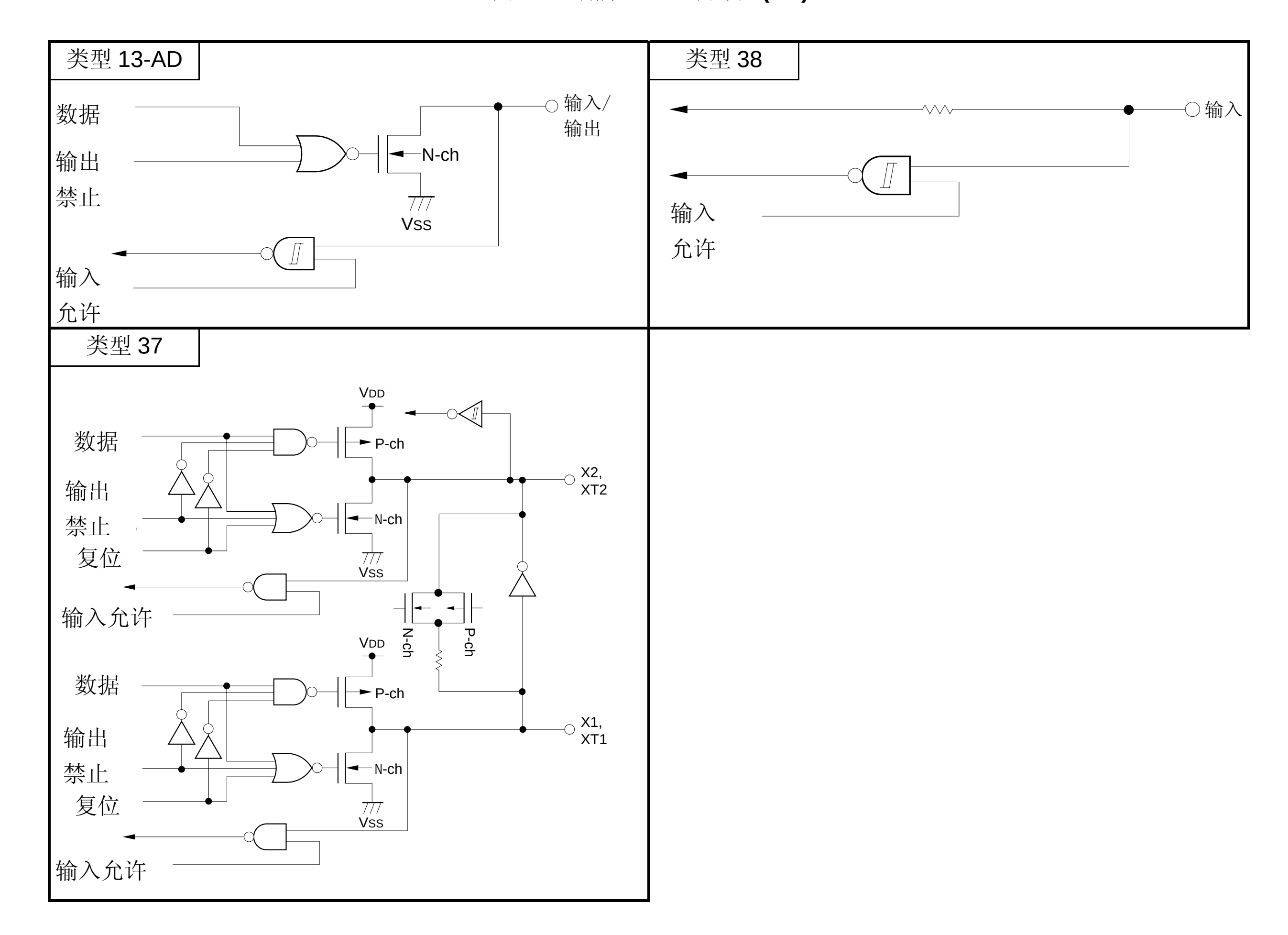


**图 2-1. 引脚 I/O 电路列表 (2/2)**



## **3.1** 存储空间

78K0/KC2 产品可以访问大小为 64KB 的存储空间。图 3-1 ～ 3-7 显示了存储空间映射图。

**注意事项　不管内部存储器容量有多少，内部存储器容量切换寄存器 (IMS) 和所有 78K0/KC2 产品的内部扩展 RAM 容量切换寄存器 (IXS) 的初始值都是固定的 (IMS = CFH, IXS = 0CH) 。因此每种产品设置值如下所示。**

表 **3-1.** 设置内存容量切换寄存器 **(IMS)** 和内部扩展 **RAM** 容量切换寄存器 **(IXS)** 的值

| Flash 存储器版本 (78K0/KC2) | IMS | IXS | ROM 容量 | 内部高速 RAM 容量 | 内部扩展 RAM 容量 |
|---|---|---|---|---|---|
| μPD78F0511 | 04H | 0CH | 16 KB | 768 字节 | – |
| μPD78F0512 | C6H | | 24 KB | 1 KB | |
| μPD78F0513, 78F0513D | C8H | | 32 KB | | |
| μPD78F0514 | CCH | 0AH | 48 KB | | 1 KB |
| μPD78F0515, 78F0515D | CFH | 08H | 60 KB | | 2 KB |

**图 3-1. 存储空间映射图 (μPD78F0511)**

FFFFH
特殊功能寄存器（SFR）256 × 8 位
FF00H
FEFFH
通用寄存器 32 × 8 位
FEE0H
FEDFH
内部高速 RAM 768 × 8 位
FC00H
FBFFH
保留
4000H
3FFFH
Flash 存储器 16384 × 8 位
0000H

数据存储器空间
程序存储器空间

3FFFH
程序区域
1FFFH
1085H
1084H
选项字节区域[注 1] 5 × 8 位
1080H
107FH
程序区域
引导簇 1
1000H
0FFFH
CALLF 入口区域 2048 × 8 位
0800H
07FFH
程序区域 1915 × 8 位
0085H
0084H
选项字节区域[注 1] 5 × 8 位
0080H
007FH
CALLT 表区域 64 × 8 位
0040H
003FH
向量表区域 64 × 8 位
0000H
引导簇 0[注 2]

**注** **1.** 当不使用引导交换功能的时候，设置选项字节区域： 0080H ～ 0084H 。
当使用引导交换功能的时候，设置选项字节区域 ： 0080H ～ 0084H 和 1080H ～ 1084H 。
**2.** 根据安全设置可以禁止写入引导簇 0 (参见 **25.8 安全设置**)。

<R> **备注** flash 存储器被分割成 block (1 block = 1 KB)。 对于地址和 block 编号，请参看 **表 3-2 Flash 存储器中地址和 block 编号的关系**。

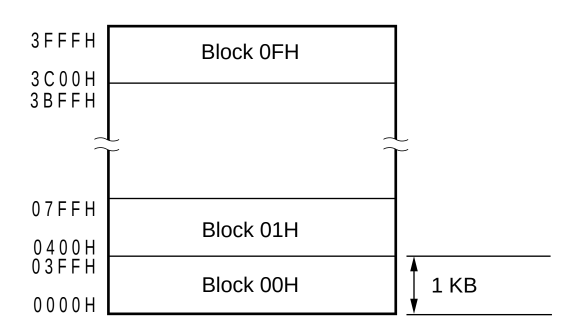


图 3-2. 存储空间映射图 ($\mu$PD78F0512)

FFFFH
特殊功能寄存器
（SFR）
256 × 8 位
FF00H
FEFFH
通用寄存器
32 × 8 位
FEE0H
FEDFH
内部高速 RAM
1024 × 8 位
FB00H
FAFFH
数据
存储器空间
保留
6000H
5FFFH
程序
存储器空间
Flash 存储器
24576 × 8 位
0000H

5FFFH
程序区域
1FFFH
1085H
1084H
选项字节区域[注1] 5 × 8 位
1080H
107FH
引导簇 1
程序区域
1000H
0FFFH
CALLF 入口区域
2048 × 8 位
0800H
07FFH
程序区域
1915 × 8 位
引导簇 0[注2]
0085H
0084H
选项字节区域[注1] 5 × 8 位
0080H
007FH
CALLT 表区域
64 × 8 位
0040H
003FH
向量表区域
64 × 8 位
0000H

注　**1.** 当不使用引导交换功能的时候，设置选项字节区域： 0080H ～ 0084H 。
当使用引导交换功能的时候，设置选项字节区域 ： 0080H ～ 0084H 和 1080H ～ 1084H 。
**2.** 根据安全设置可以禁止写入引导簇 0 (参见 **25.8** 安全设置)。

<R>　**备注**　flash 存储器被分割成 block (1 block = 1 KB)。 对于地址和 block 编号，请参看 表 **3-2 Flash** 存储器中地址和 **block** 编号的关系。

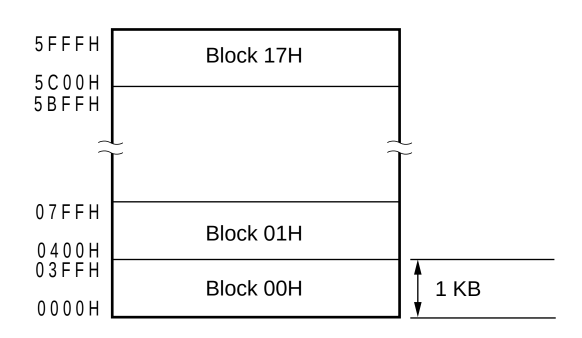


图 **3-3.** 存储空间映射图 **($\mu$PD78F0513)**

FFFFH
特殊功能寄存器
（SFR）
256 × 8 位
FF00H
FEFFH
通用寄存器
32 × 8 位
FEE0H
FEDFH
内部高速 RAM
1024 × 8 位
FB00H
FAFFH
数据
存储器空间
保留
8000H
7FFFH
程序
存储器空间
Flash 存储器
32768 × 8 位
0000H

7FFFH
程序区域
1FFFH
1085H
1084H
选项字节区域[注1] 5 × 8 位
1080H
107FH
程序区域
引导簇 1
1000H
0FFFH
CALLF 入口区域
2048 × 8 位
0800H
07FFH
程序区域
1915 × 8 位
0085H
0084H
选项字节区域[注1] 5 × 8 位
引导簇 0[注2]
0080H
007FH
CALLT 表区域
64 × 8 位
0040H
003FH
向量表区域
64 × 8 位
0000H

注 **1.** 当不使用引导交换功能的时候，设置选项字节区域： 0080H ～ 0084H 。
当使用引导交换功能的时候，设置选项字节区域 ： 0080H ～ 0084H 和 1080H ～ 1084H 。
**2.** 根据安全设置可以禁止写入引导簇 0 (参见 **25.8** 安全设置)。

<R> 备注 flash 存储器被分割成 block (1 block = 1 KB)。 对于地址和 block 编号，请参看 表 **3-2 Flash** 存储器中地址和 **block** 编号的关系。

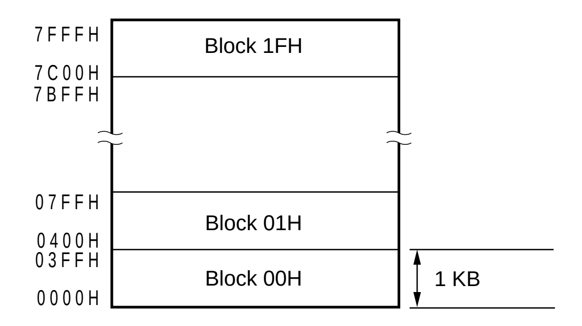


**图 3-4. 存储空间映射图 ($\mu$PD78F0513D)**

FFFFH
特殊功能寄存器（SFR）256 × 8 位
FF00H
FEFFH
通用寄存器 32 × 8 位
FEE0H
FEDFH
内部高速 RAM 1024 × 8 位
FB00H
FAFFH
保留
8000H
7FFFH
Flash 存储器 32768 × 8 位
0000H

数据存储器空间
程序存储器空间

7FFFH
程序区域
1FFFH
108FH
108EH
片上调试安全 ID 设置区域 10×8 位
1085H
1084H
选项字节区域[注 1] 5 × 8 位
1080H
107FH
程序区域
1000H
引导簇 1
0FFFH
CALLF 入口区域 2048 × 8 位
0800H
07FFH
程序区域 1915 × 8 位
008FH
008EH
片上调试安全 ID 设置区域 10×8 位
0085H
0084H
选项字节区域[注 1] 5 X 8 位
0080H
007FH
CALLT 表区域 64 × 8 位
0040H
003FH
向量表区域 64 × 8 位
0000H
引导簇 0[注 2]

注　**1.** 当不使用引导交换功能的时候，设置选项字节区域： 0080H ～ 0084H，并且片上调试安全 ID 区域：0085H ～ 008EH。
当使用引导交换功能的时候，设置选项字节区域 ： 0080H ～ 0084H 和 1080H ～ 1084H ，并且片上调试安全 ID 区域：0085H ～ 008EH 和 1085H ～ 108EH。
**2.** 根据安全设置可以禁止写入引导簇 0 (参见 **25.8 安全设置**)。

<R>　备注　flash 存储器被分割成 block (1 block = 1 KB)。 对于地址和 block 编号，请参看 **表 3-2 Flash 存储器中地址和 block 编号的关系。**

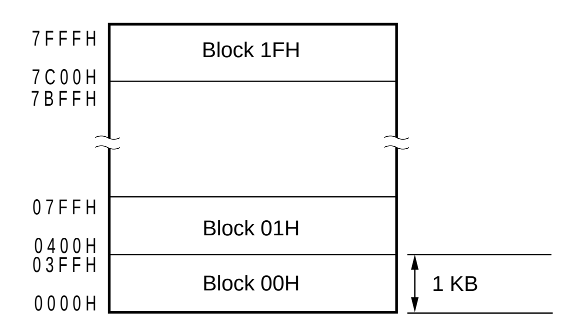


**图 3-5. 存储空间映射图 ($\mu$PD78F0514)**

FFFFH
特殊功能寄存器（SFR）256 × 8 位
FF00H
FEFFH
通用寄存器 32 × 8 位
FEE0H
FEDFH
内部高速 RAM 1024 × 8 位
FB00H
FAFFH
保留
F800H
F7FFH
内部扩展 RAM 1024 × 8 位
F400H
F3FFH
保留
C000H
BFFFH
Flash 存储器 49152 × 8 位
0000H

数据存储器空间

程序 RAM 区域可获取指令的 RAM 空间

程序存储器空间

BFFFH
程序区域
1085H
1084H
选项字节区域[注 1] 5 × 8 位
1080H
107FH
程序区域
1000H
0FFFH
CALLF 入口区域 2048 × 8 位
0800H
07FFH
程序区域 1915 × 8 位
0085H
0084H
选项字节区域[注 1] 5 × 8 位
0080H
007FH
CALLT 表区域 64 × 8 位
0040H
003FH
向量表区域 64 × 8 位
0000H

1FFFH
引导簇 1
引导簇 0 [注 2]

注 **1.** 当不使用引导交换功能的时候，设置选项字节区域： 0080H ～ 0084H 。
当使用引导交换功能的时候，设置选项字节区域 ： 0080H ～ 0084H 和 1080H ～ 1084H 。
**2.** 根据安全设置可以禁止写入引导簇 0 (参见 **25.8 安全设置**)。

<R> **备注** flash 存储器被分割成 block (1 block = 1 KB)。 对于地址和 block 编号，请参看 表 **3-2 Flash 存储器中地址和 block 编号的关系**。

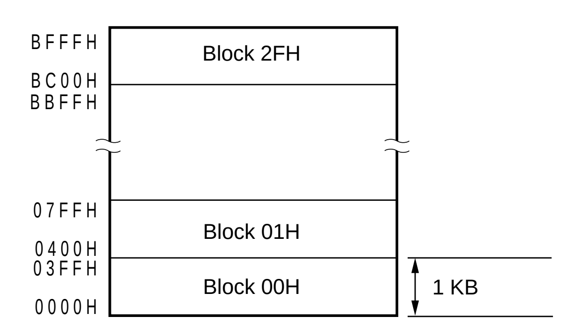


**图 3-6. 存储空间映射图 (μPD78F0515)**

FFFFH
特殊功能寄存器
（SFR）
256 × 8 位
FF00H
FEFFH
通用寄存器
32 × 8 位
FEE0H
FEDFH
内部高速 RAM
1024 × 8 位
FB00H
FAFFH
保留
数据存储
器空间
F800H
F7FFH
程序 RAM 区域
可获取指令的
RAM 空间
内部扩展 RAM
2048 × 8 位
F000H
EFFFH
Flash 存储器
61440 × 8 位
程序存储
器空间
0000H

EFFFH
程序区域
1FFFH
1085H
1084H
选项字节区域[注1] 5 × 8 位
1080H
107FH
引导簇 1
程序区域
1000H
0FFFH
CALLF 入口区域
2048 × 8 位
0800H
07FFH
程序区域
1915 × 8 位
引导簇 0[注2]
0085H
0084H
选项字节区域[注1] 5 X 8 位
0080H
007FH
CALLT 表区域
64 × 8 位
0040H
003FH
向量表区域
64 × 8 位
0000H

注　**1.** 当不使用引导交换功能的时候，设置选项字节区域： 0080H ～ 0084H 。
当使用引导交换功能的时候，设置选项字节区域 ： 0080H ～ 0084H 和 1080H ～ 1084H 。
**2.** 根据安全设置可以禁止写入引导簇 0 (参见 **25.8** **安全设置**)。

<R>　备注　flash 存储器被分割成 block (1 block = 1 KB)。 对于地址和 block 编号，请参看 **表 3-2 Flash 存储器中地址和 block 编号的关系**。

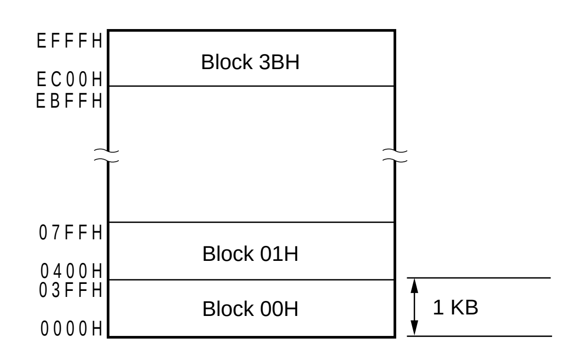


**图 3-7. 存储空间映射图 ($\mu$PD78F0515D)**

FFFFH
特殊功能寄存器
（SFR）
256 × 8 位
FF00H
FEFFH
通用寄存器
32 × 8 位
FEE0H
FEDFH
内部高速 RAM
1024 × 8 位
FB00H
FAFFH
保留
F800H
F7FFH
内部扩展 RAM
2048 × 8 位
F000H
EFFFH
Flash 存储器
61440 × 8 位
0000H

数据存储器空间
程序 RAM 区域
可获取指令的 RAM 空间
程序存储器空间

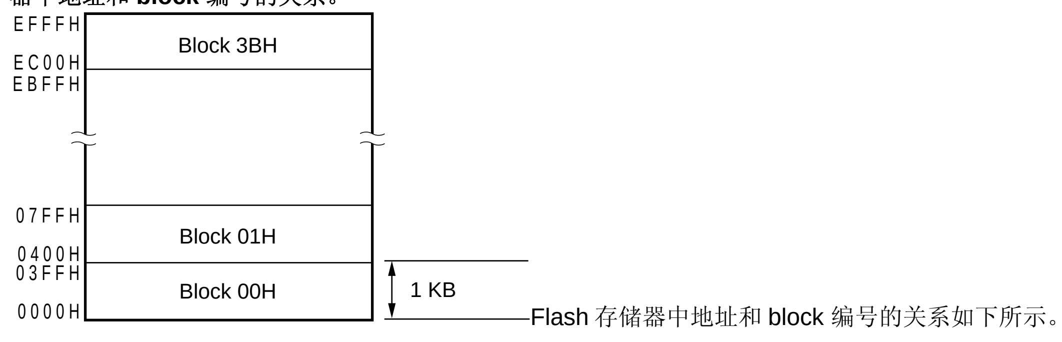
EFFFH
程序区域
108FH
108EH
片上调试安全 ID 设置区域[注 1]
10 × 8 位
1085H
1084H
选项字节区域[注 1] 5 × 8 位
1080H
107FH
程序区域
1000H
0FFFH
CALLF 入口区域
2048 × 8 位
0800H
07FFH
程序区域
1905 × 8 位
008FH
008EH
片上调试安全 ID 设置区域[注 1]
10 × 8 位
0085H
0084H
选项字节区域[注 1] 5 X 8 位
0080H
007FH
CALLT 表区域
64 × 8 位
0040H
003FH
向量表区域
64 × 8 位
0000H

1FFFH
引导簇 1
引导簇 0 [注 2]

注 **1.** 当不使用引导交换功能的时候，设置选项字节区域： 0080H ～ 0084H，并且片上调试安全 ID 区域：0085H ～ 008EH。
当使用引导交换功能的时候，设置选项字节区域 ： 0080H ～ 0084H 和 1080H ～ 1084H ，并且片上调试安全 ID 区域：0085H ～ 008EH 和 1085H ～ 108EH。

**2.** 根据安全设置可以禁止写入引导簇 0 (参见 **25.8 安全设置**)。

<R> **备注** flash 存储器被分割成 block (1 block = 1 KB)。 对于地址和 block 编号，请参看 **表 3-2 Flash 存储器中地址和 block 编号的关系**。




Flash 存储器中地址和 block 编号的关系如下所示。

<R> **表 3-2. Flash 存储器中地址和 block 编号的关系**

| 地址值 | Block 编号 | 地址值 | Block 编号 | 地址值 | Block 编号 | 地址值 | Block 编号 |
|---|---|---|---|---|---|---|---|
| 0000H ～ 03FFH | 00H | 4000H ～ 43FFH | 10H | 8000H ～ 83FFH | 20H | C000H ～ C3FFH | 30H |
| 0400H ～ 07FFH | 01H | 4400H ～ 47FFH | 11H | 8400H ～ 87FFH | 21H | C400H ～ C7FFH | 31H |
| 0800H ～ 0BFFH | 02H | 4800H ～ 4BFFH | 12H | 8800H ～ 8BFFH | 22H | C800H ～ CBFFH | 32H |
| 0C00H ～ 0FFFH | 03H | 4C00H ～ 4FFFH | 13H | 8C00H ～ 8FFFH | 23H | CC00H ～ CFFFH | 33H |
| 1000H ～ 13FFH | 04H | 5000H ～ 53FFH | 14H | 9000H ～ 93FFH | 24H | D000H ～ D3FFH | 34H |
| 1400H ～ 17FFH | 05H | 5400H ～ 57FFH | 15H | 9400H ～ 97FFH | 25H | D400H ～ D7FFH | 35H |
| 1800H ～ 1BFFH | 06H | 5800H ～ 5BFFH | 16H | 9800H ～ 9BFFH | 26H | D800H ～ DBFFH | 36H |
| 1C00H ～ 1FFFH | 07H | 5C00H ～ 5FFFH | 17H | 9C00H ～ 9FFFH | 27H | DC00H ～ DFFFH | 37H |
| 2000H ～ 23FFH | 08H | 6000H ～ 63FFH | 18H | A000H ～ A3FFH | 28H | E000H ～ E3FFH | 38H |
| 2400H ～ 27FFH | 09H | 6400H ～ 67FFH | 19H | A400H ～ A7FFH | 29H | E400H ～ E7FFH | 39H |
| 2800H ～ 2BFFH | 0AH | 6800H ～ 6BFFH | 1AH | A800H ～ ABFFH | 2AH | E800H ～ EBFFH | 3AH |
| 2C00H ～ 2FFFH | 0BH | 6C00H ～ 6FFFH | 1BH | AC00H ～ AFFFH | 2BH | EC00H ～ EFFFH | 3BH |
| 3000H ～ 33FFH | 0CH | 7000H ～ 73FFH | 1CH | B000H ～ B3FFH | 2CH | | |
| 3400H ～ 37FFH | 0DH | 7400H ～ 77FFH | 1DH | B400H ～ B7FFH | 2DH | | |
| 3800H ～ 3BFFH | 0EH | 7800H ～ 7BFFH | 1EH | B800H ～ BBFFH | 2EH | | |
| 3C00H ～ 3FFFH | 0FH | 7C00H ～ 7FFFH | 1FH | BC00H ～ BFFFH | 2FH | | |

备注 μPD78F0511: Block 编号 00H ～ 0FH
μPD78F0512: Block 编号 00H ～ 17H
μPD78F0513, 78F0513D: Block 编号 00H ～ 1FH
μPD78F0514: Block 编号 00H ～ 2FH
μPD78F0515, 78F0515D: Block 编号 00H ～ 3BH

### 3.1.1 内部程序存储空间

内部程序存储空间用于存储程序和表数据，一般通过程序计数器(PC)来寻址。
78K0/KC2 产品内部 ROM(Flash 存储器)的情况如下表所示。

**表 3-3. 内部 ROM 空间**

| 型号 | 内部 ROM | |
|---|---|---|
| | 结构 | 容量 |
| μPD78F0511 | Flash 存储器 | 16384 × 8 位 (0000H ～ 3FFFH) |
| μPD78F0512 | | 24576 × 8 位 (0000H ～ 5FFFH) |
| μPD78F0513, 78F0513D | | 32768 × 8 位 (0000H ～ 7FFFH) |
| μPD78F0514 | | 49152 × 8 位 (0000H ～ BFFFH) |
| μPD78F0515, 78F0515D | | 61440 × 8 位 (0000H ～ EFFFH) |

内部程序存储空间主要分为以下几个区域。

**(1) 向量表区域**

从 0000H 到 003FH 总共 64 字节作为向量表区域。在向量表中存放的是根据复位信号输入或每个中断请求的产生进行转移的程序的起始地址。
在 16 位地址中，低 8 位是偶地址，高 8 位是奇地址。

**表 3-4. 向量表**

| 向量表地址 | 中断源 | 向量表地址 | 中断源 |
|---|---|---|---|
| 0000H | $\overline{\text{RESET}}$ 输入, POC, LVI, WDT | 001CH | INTTMH0 |
| 0004H | INTLVI | 001EH | INTTM50 |
| 0006H | INTP0 | 0020H | INTTM000 |
| 0008H | INTP1 | 0022H | INTTM010 |
| 000AH | INTP2 | 0024H | INTAD |
| 000CH | INTP3 | 0026H | INTSR0 |
| 000EH | INTP4 | 0028H | INTWTI |
| 0010H | INTP5 | 002AH | INTTM51 |
| 0012H | INTSRE6 | 002CH | INTKR |
| 0014H | INTSR6 | 002EH | INTWT |
| 0016H | INTST6 | 0030H | INTP6[注 1] |
| 0018H | INTCSI10/INTST0 | 0034H | INTIIC0/INTDMU[注 2] |
| 001AH | INTTMH1 | 003EH | BRK |

注 **1.** 只在 48 引脚产品可用。
  **2.** 只在μPD78F0514, 78F0515, 和 78F0515D 可用。

**(2) CALLT 指令表区域**

0040H 至 007FH 共 64 字节的区域，可存放 1 字节调用指令(CALLT)的子程序入口地址。

**(3)** 选项字节区域

0080H ～ 0084H 和 1080H ～ 1084H 各 5 字节的区域可以用作选项字节区域。当不使用引导交换功能时，在 0080H ～ 0084H 设置选项字节；而当使用引导交换功能时，在 0080H ～ 0084H 和 1080H ～ 1084 H 设置选项字节。详细信息参见 **第二十四章 选项字节**。

**(4) CALLF 指令入口区域**

0800H ～ 0FFFH 的区域，用于子程序的直接调用，通过一个 2 字节的调用指令(CALLF)实现。

**(5)** 片上调试安全 **ID** 设置区域 **(**仅用于$\mu$**PD78F0513D** 和 **78F0515D )**

0085H ～ 008EH 和 1085H ～ 108EH 各 10 字节区域可用作片上调试安全 ID 设置区域。不使用引导交换功能时，在 0085H ～ 008EH 区域设置片上调试安全 ID；而当使用引导交换功能时，在 0085H～ 008EH 和 1085H ～ 108EH 区域设置。详细情况请参见 **第二十六章 片上调试功能 (**仅用于$\mu$**PD78F0513D** 和 **78F0515D** )。

### 3.1.2 内部数据存储空间

78K0/KC2 产品包括以下几种 RAM。

**(1) 内部高速 RAM**

**表 3-5. 内部高速 RAM 容量**

| 型号 | 内部高速 RAM |
|---|---|
| μPD78F0511 | 768 × 8 位 (FC00H ～ FEFFH) |
| μPD78F0512 | 1024 × 8 位 (FB00H ～ FEFFH) |
| μPD78F0513, 78F0513D | |
| μPD78F0514 | |
| μPD78F0515, 78F0515D | |

FEE0H ～ FEFFH 共 32 字节的区域分配给 4 个通用寄存器 bank，每个 bank 都由 8 个 8 位寄存器组成。
这个区域不可以用作写和执行指令的程序区域。
内部高速 RAM 还可以作为堆栈存储器使用。

**(2) 内部扩展 RAM**

**表 3-6. 内部扩展 RAM 容量**

| 型号 | 内部扩展 RAM |
|---|---|
| μPD78F0511 | – |
| μPD78F0512 | |
| μPD78F0513, 78F0513D | |
| μPD78F0514 | 1024 × 8 位 (F400H ～ F7FFH) |
| μPD78F0515, 78F0515D | 2048 × 8 位 (F000H ～ F7FFH) |

与内部高速 RAM 类似，内部扩展 RAM 可作为公共数据区域，同时也可作为写和执行指令的程序存储区域。
内部扩展 RAM 不能用作堆栈寄存器。

### 3.1.3 特殊功能寄存器(SFR)区域

片内外围硬件的特殊功能寄存器 (SFR)被分配在 FF00H ～ FFFFH 的区域 (参见 **3.2.3 特殊功能寄存器(SFR)**中表 **3-7 特殊功能寄存器列表)**。

**注意事项 不要访问那些未分配特殊功能寄存器的地址区域。**

### 3.1.4 数据存储空间寻址

寻址是定位地址的方式，需要定位的地址包括下一条指令地址或者与指令执行相关的存储器地址或寄存器地址。

基于可操作性和其他考虑，在 78K0/KC2 中提供了几种用于与指令执行相关的存储器寻址方式。因为有专门的数据存储区域，故可使用一些特殊的寻址方式，具有特殊功能寄存器(SFR)和通用寄存器的功能。图 3-8 到 3-14 显示了数据存储空间与寻址方式的对应关系。如需了解每种寻址方式的详细内容，参见 **3.4 操作数地址寻址**。

图 3-8. 数据存储空间与寻址方式的对应关系 (μPD78F0511)

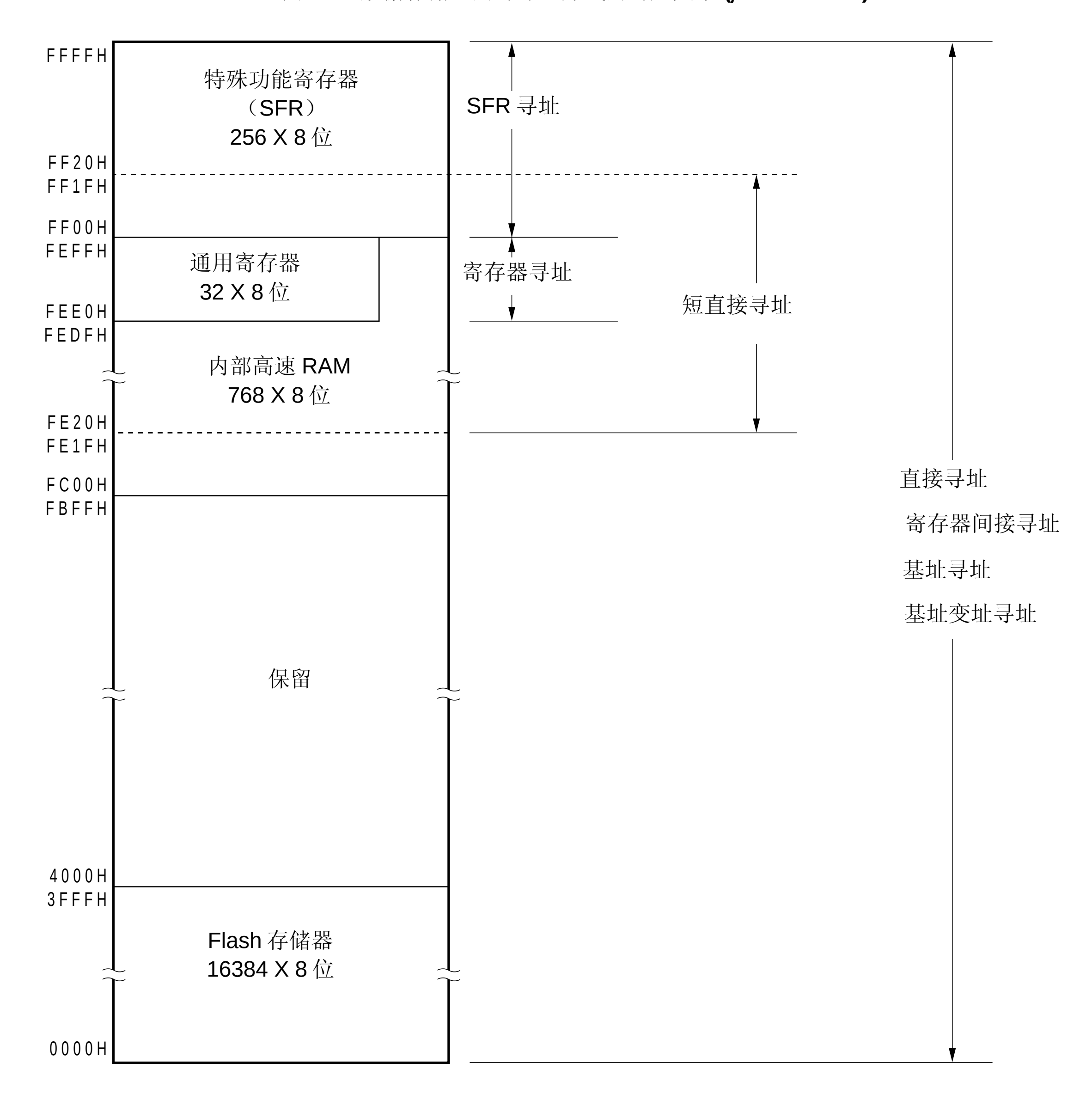


**图 3-9. 数据存储空间与寻址方式的对应关系 (μPD78F0512)**

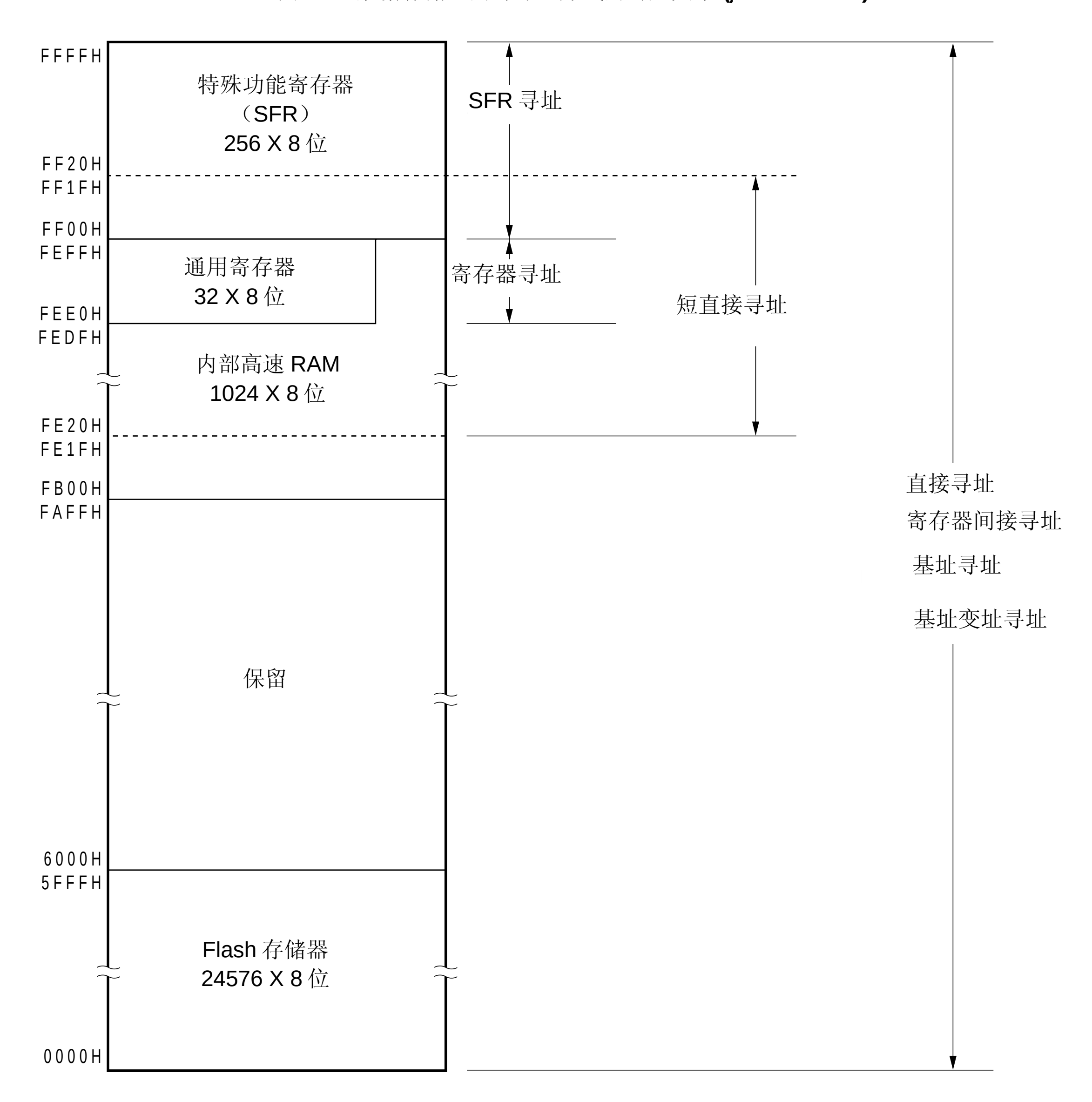


**图 3-10. 数据存储空间与寻址方式的对应关系 (μPD78F0513 和 78F0513D)**

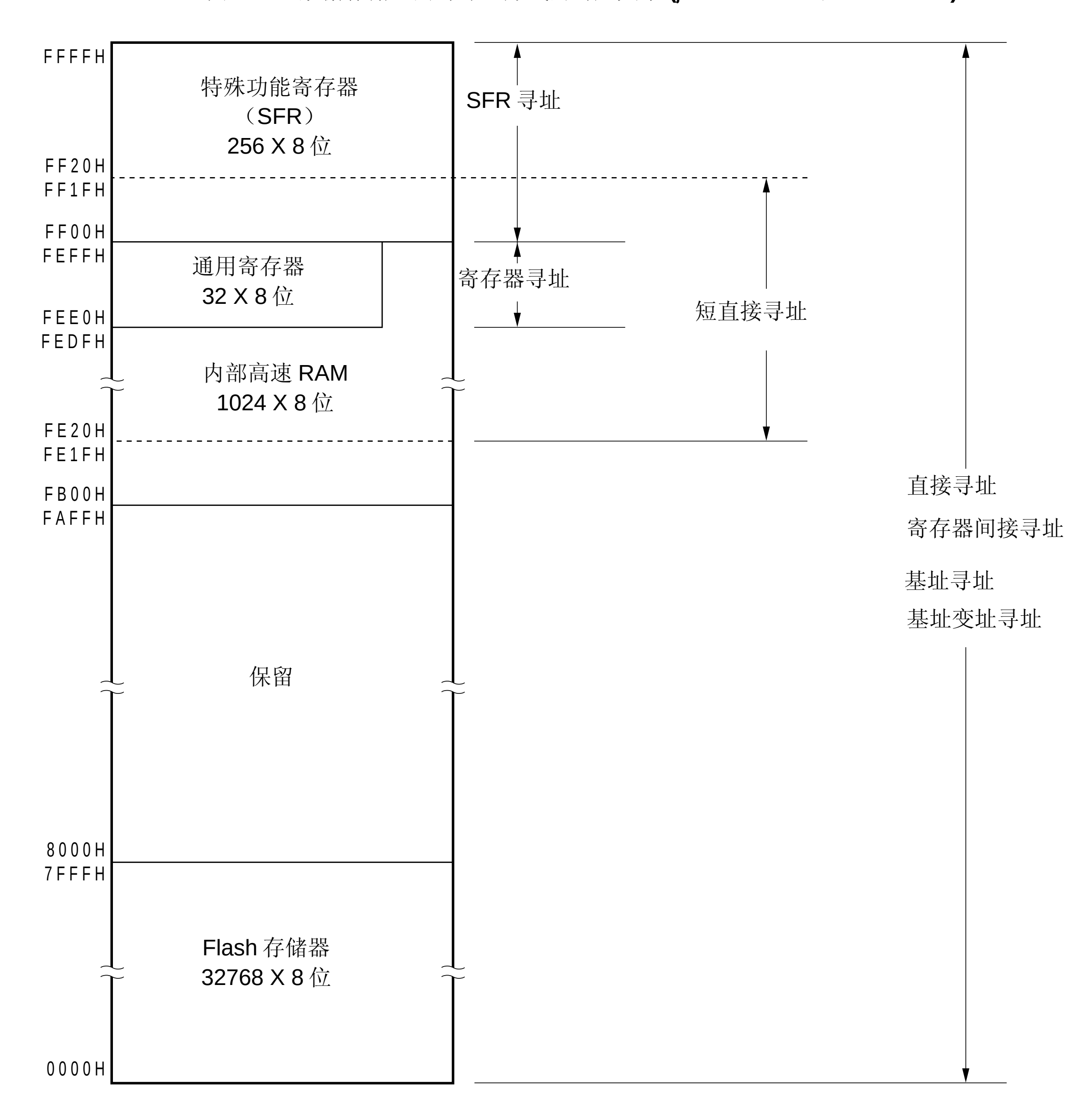


图 **3-11 .** 数据存储空间与寻址方式的对应关系 **(*μ*PD78F0514)**

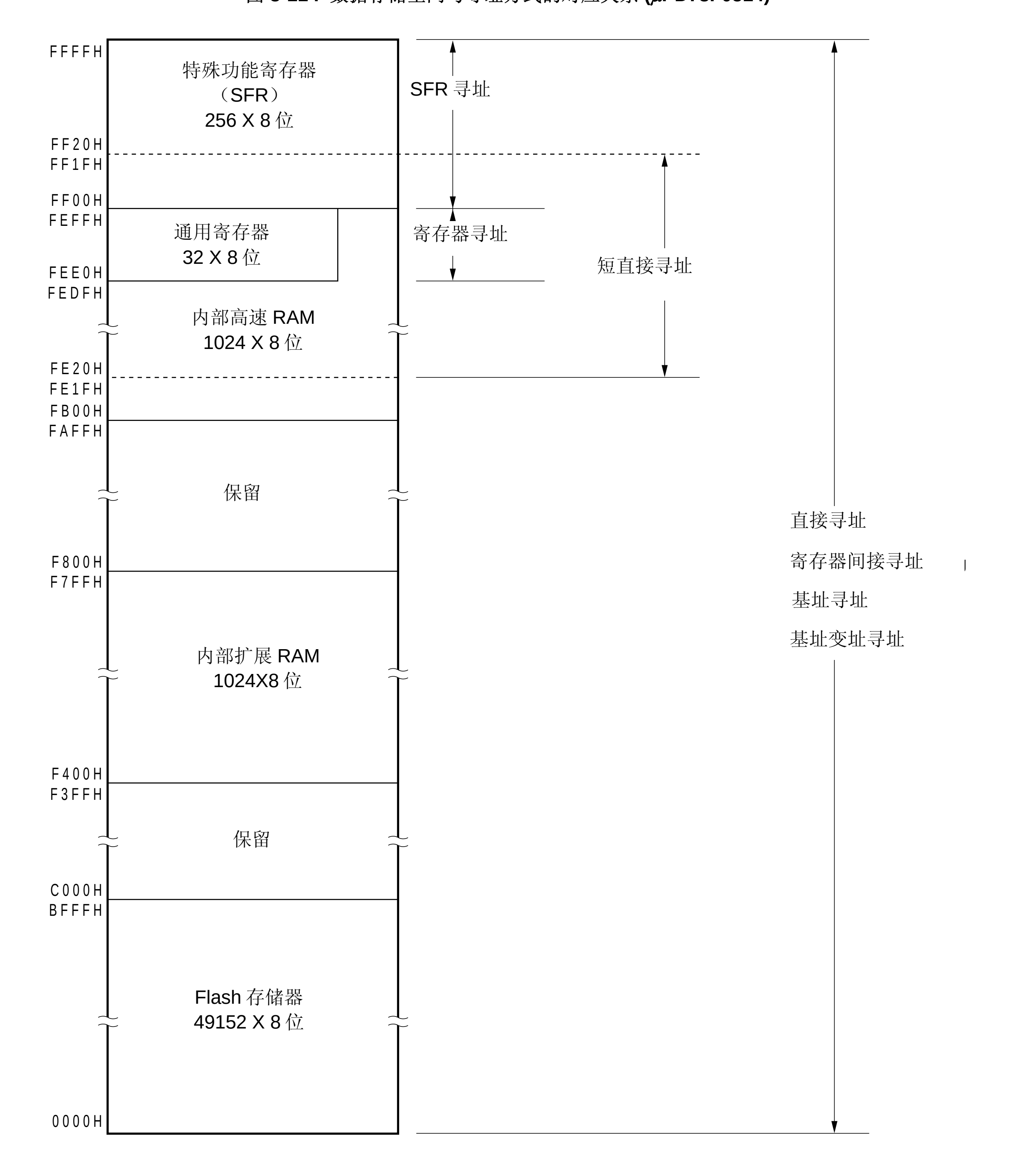
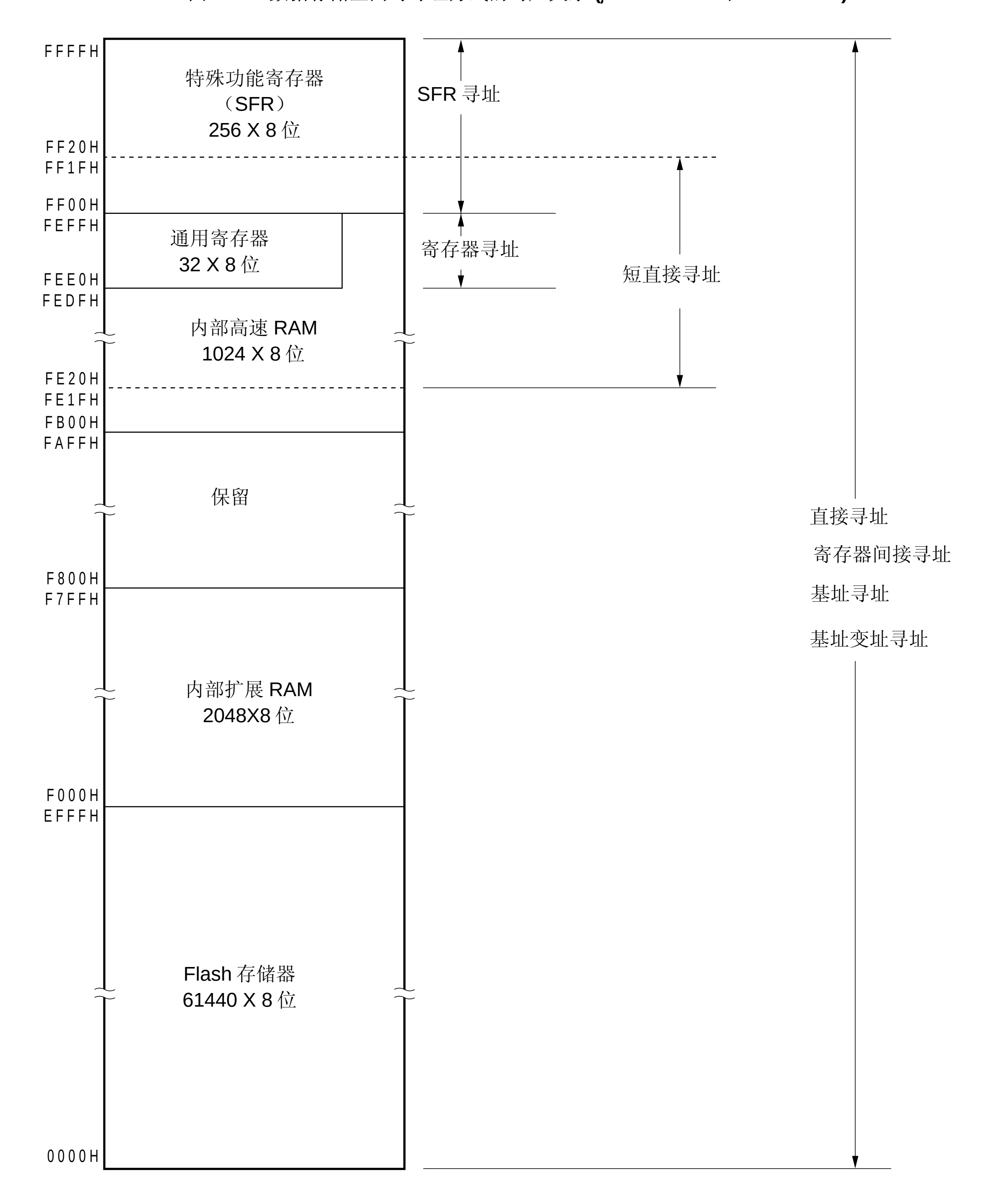


**图 3-12. 数据存储空间与寻址方式的对应关系 ($\mu$PD78F0515 和 78F0515D)**



## 3.2 处理器寄存器

78K0/KC2 产品包含以下几种处理器寄存器。

### 3.2.1 控制寄存器

控制寄存器用于控制程序执行的顺序、状态和堆栈空间。程序计数器(PC)、程序状态字(PSW)和堆栈指针寄存器(SP)都属于控制寄存器。

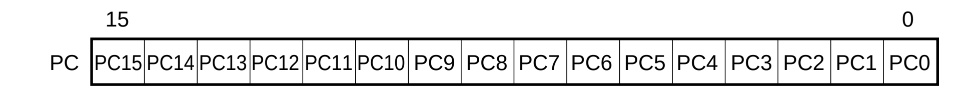
**(1) 程序计数器 (PC)**

程序计数器是一个 16 位寄存器，用于存放下一条即将要执行的指令的地址。

在正常情况下，根据获取的指令字节数，程序计数器(PC)的值会自动累加。当执行分支指令时，则设置立即数和寄存器内容。

复位信号的产生将复位向量表中地址为 0000H 和 0001H 中的值赋给程序计数器。

**图 3-13. 程序计数器格式**



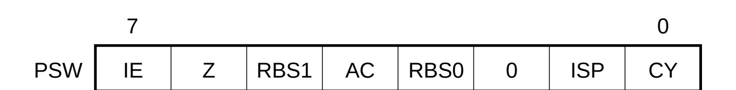

**(2) 程序状态字 (PSW)**

程序状态字(PSW)是一个 8 位寄存器，由各种标志位组成，通过指令执行对其进行设置或复位。

根据中断请求的产生或 PUSH PSW 指令执行，程序状态字的内容自动入栈；通过执行 RETB,RETI 和 POP PSW 指令，程序状态字的值自动恢复。

复位信号的产生将程序状态字的内容设置为 02H。

**图 3-14. 程序状态字格式**




**(a) 中断允许标志 (IE)**

该标志用于控制 CPU 响应中断请求操作。

当 IE 为 0 时，表示不允许中断(DI)，即禁止所有可屏蔽中断请求。

当 IE 为 1 时，表示允许中断(EI)，通过优先服务标志(ISP)、用于各种中断源的中断屏蔽标志以及优先级规定标志来完成响应中断请求的控制。

当执行 DI 指令或中断请求得到响应时，该标志复位(0)；当执行 EI 指令时，该标志设置为 1。

**(b) 零标志 (Z)**

当操作结果为 0 时，该标志置 1，其他情况置 0。

**(c) 寄存器组选择标志 (RBS0 和 RBS1)**

寄存器组选择标志有两位，用于选择四组寄存器中的一组。
标志位中存储的信息用来指明执行 SEL RBn 指令时所选择的寄存器组。

**(d) 半进位标志 (AC)**

如果操作结果中第 3 位有进位或在第 3 位上有借位，则该标志置 1。其他情况该标志置 0。

**(e) 优先服务标志 (ISP)**

该标志用来管理可屏蔽向量中断响应的优先级。当 ISP 为 0 时，由优先级指定标志寄存器(PR0L, PR0H, PR1L, PR1H) (参见 **18.3 (3) 优先级指定标志寄存器(PR0L, PR0H, PR1L, PR1H)**)指定为低优先级的向量中断请求被禁止响应。对请求的实际响应是由中断允许标志(IE)的状态控制的。

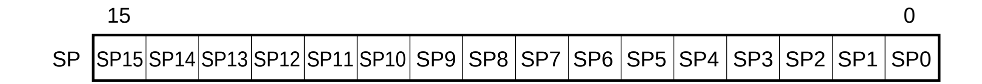
**(f) 进位标志 (CY)**

该标志存储的是在执行加减指令时出现的进位或借位。它也存储循环指令执行中的转移值，还可以在位操作指令执行中作为位累加器使用。

**(3) 堆栈指针 (SP)**

这是一个 16 位的寄存器，用来存放存储器堆栈区的起始地址。只有内部高速 RAM 区域才能被设置为堆栈区。

**图 3-15. 堆栈指针格式**

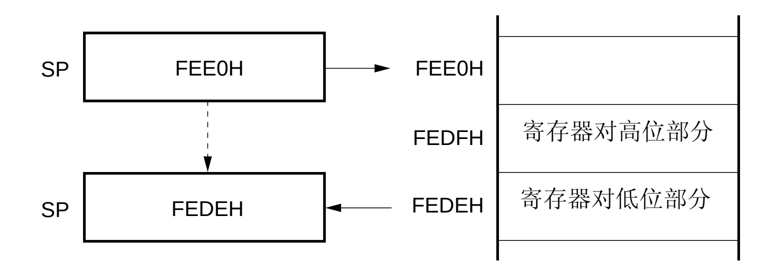



在向堆栈写(存)数据时，堆栈指针 SP 递减，而从堆栈中读出(恢复)数据时，堆栈指针累加。
堆栈的数据存储/恢复操作过程如图 3-16 和 3-17 所示。

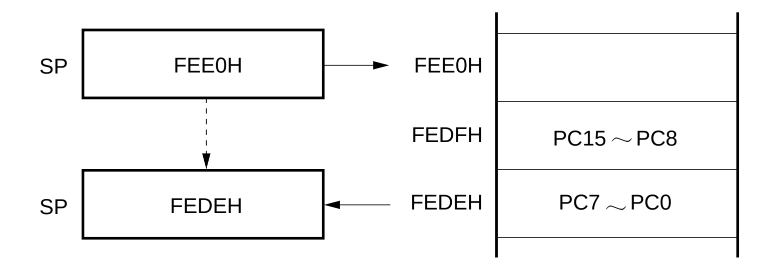
**注意事项 由于复位信号产生时，SP 的内容不确定，所以在使用堆栈前必须先对 SP 初始化。**

图 3-16. 将数据存入堆栈

**(a) PUSH rp 指令 (当 SP = FEE0H)**




**(b) CALL, CALLF, CALLT 指令 (当 SP = FEE0H)**




**(c) 中断, BRK 指令 (当 SP = FEE0H)**

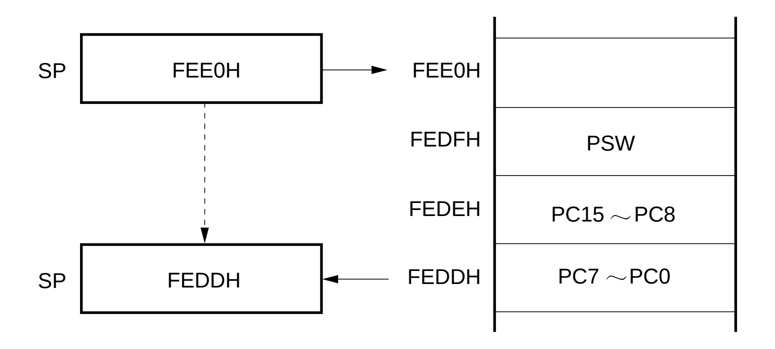
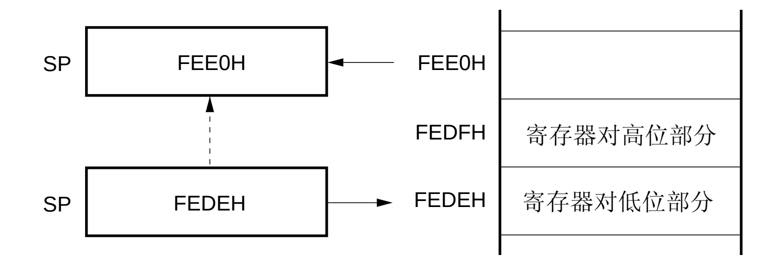
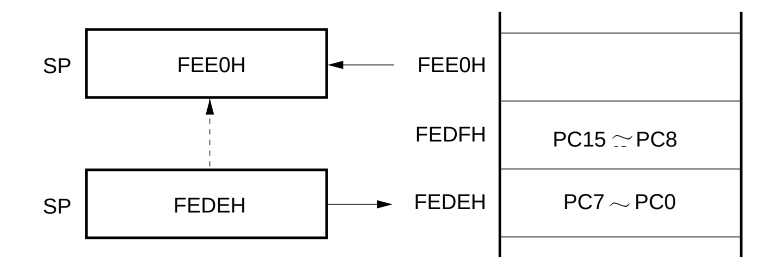


图 **3-17.** 从堆栈读出数据

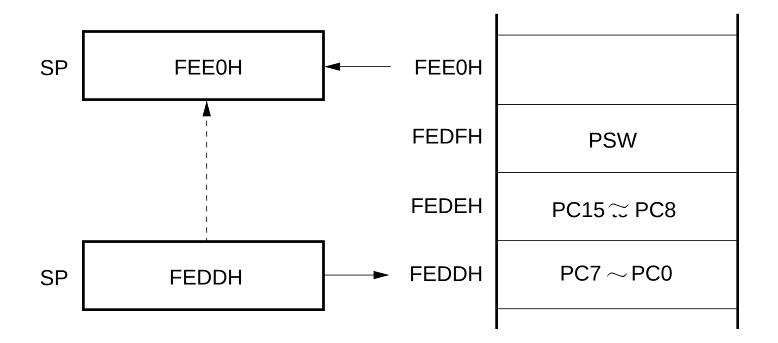
**(a) POP rp 指令 (当 SP = FEDEH)**




**(b) RET 指令 (当 SP = FEDEH)**




**(c) RETI, RETB 指令 (当 SP = FEDDH)**



**3.2.2 通用寄存器**

通用寄存器映射到数据存储器特定的地址空间为 FEE0H ～ FEFFH。通用寄存器共有四个 bank，每一个 bank 由 8 个 8 位寄存器(X, A, C, B, E, D, L 和 H)组成。

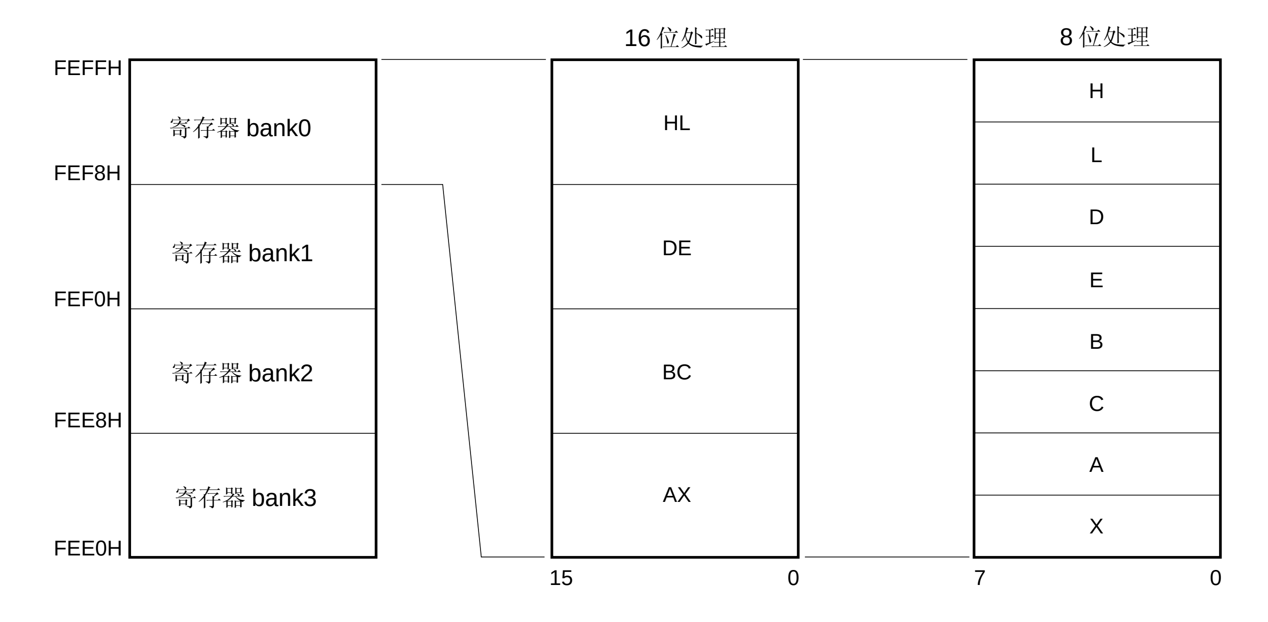
每个寄存器可作为一个 8 位寄存器使用，两个成对的 8 位寄存器可作为一个 16 位寄存器(AX, BC, DE 和 HL)使用。

描述通用寄存器时，可以使用功能名称(X, A, C, B, E, D, L, H, AX, BC, DE 和 HL)或绝对名称(R0 ～ R7，RP0 ～ RP3)。

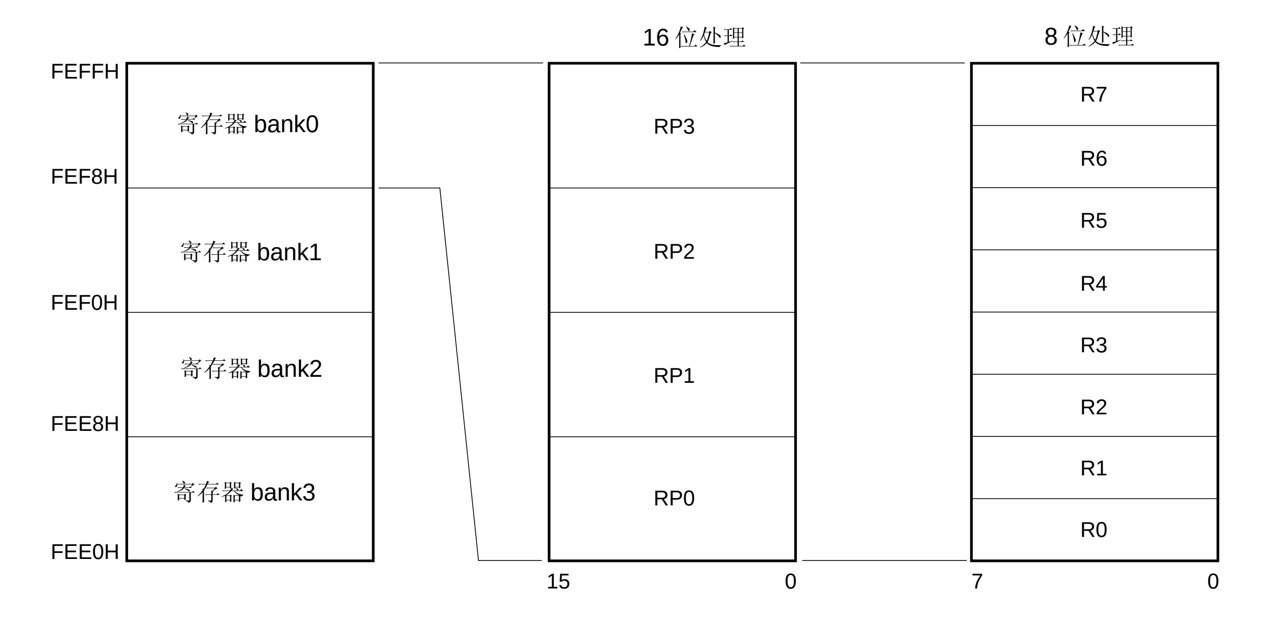
用于指令执行的寄存器 bank 由 CPU 控制指令(SEL RBn)来设置。由于 4 个寄存器 bank 的结构，通过一个用于正常处理的寄存器和另一个用于中断处理的寄存器之间的切换，可以创建一个高效率的程序。

**图 3-18. 通用寄存器结构**

**(a) 功能名称**




**(b) 绝对名称**



### 3.2.3 特殊功能寄存器 (SFRs)

与通用寄存器不同，每个特殊功能寄存器都有特定的功能。

特殊寄存器指定在FF00H到FFFFH区域。

特殊功能寄存器可像通用寄存器那样用运算指令、传送指令以及位操作指令进行操作。根据特殊功能寄存器的类型不同，可操作的位单元也不同，可以是1位、8位和16位。

每种位单元操作的描述如下。

- 1 位操作
  1 位操作指令的操作数 (sfr.bit) 被描述为汇编程序的保留符号。
  该操作也可由一个地址来定义。
- 8 位操作
  8 位操作指令的操作数 (sfr) 被描述为汇编程序的保留符号。
  该操作也可由一个地址来定义。
- 16 位操作
  16 位操作指令的操作数 (sfrp) 被描述为汇编程序的保留符号。
  寻址时表示为一个偶地址。

表 3-7 为特殊功能寄存器列表。表中术语的含义如下。

- 符号
  符号表示特殊功能寄存器的地址。它在 RA78K0 中是保留字，并使用 CC78K0 中的#pragma sfr 指令定义为一个 sfr 变量。在使用 RA78K0，ID78K0-QB 和 78K0/KX2 的 SM+时，可以将符号作为指令操作数执行写操作。
- R/W
  表示特殊功能寄存器可读或可写。
  R/W: 可读/写
  R: 只读
  W: 只写
- 可操作的位单元
  表示可操作的位单元 (1, 8, 或 16)。 “–” 表示不可操作的位单元 。
- 复位后
  表示复位信号产生后每个寄存器的状态

表 3-7. 特殊功能寄存器列表 (1/4)

| 地址 | 特殊功能寄存器(SFR)名称 | | 符号 | R/W | 可操作位单元 | | | 复位后 |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 位 | 8 位 | 16 位 | |
| FF00H | 端口寄存器 0 | | P0 | R/W | √ | √ | – | 00H |
| FF01H | 端口寄存器 1 | | P1 | R/W | √ | √ | – | 00H |
| FF02H | 端口寄存器 2 | | P2 | R/W | √ | √ | – | 00H |
| FF03H | 端口寄存器 3 | | P3 | R/W | √ | √ | – | 00H |
| FF04H | 端口寄存器 4 | | P4 | R/W | √ | √ | – | 00H |
| FF06H | 端口寄存器 6 | | P6 | R/W | √ | √ | – | 00H |
| FF07H | 端口寄存器 7 | | P7 | R/W | √ | √ | – | 00H |
| FF08H | 10 位 A/D 转换结果寄存器 | | ADCR | R | – | – | √ | 0000H |
| FF09H | | 8 位 A/D 转换结果寄存器 | ADCRH | R | – | √ | – | 00H |
| FF0AH | 接收缓冲寄存器 6 | | RXB6 | R | – | √ | – | FFH |
| FF0BH | 发送缓冲寄存器 6 | | TXB6 | R/W | – | √ | – | FFH |
| FF0CH | 端口寄存器 12 | | P12 | R/W | √ | √ | – | 00H |
| FF0DH | 端口寄存器 13[注] | | P13 | R/W | √ | √ | – | 00H |
| FF0EH | 端口寄存器 14[注] | | P14 | R/W | √ | √ | – | 00H |
| FF0FH | 串行 I/O 移位寄存器 10 | | SIO10 | R | – | √ | – | 00H |
| FF10H | 16 位定时器计数器 00 | | TM00 | R | – | – | √ | 0000H |
| FF11H | | | | | | | | |
| FF12H | 16 位定时器捕获/比较寄存器 000 | | CR000 | R/W | – | – | √ | 0000H |
| FF13H | | | | | | | | |
| FF14H | 16 位定时器捕获/比较寄存器 010 | | CR010 | R/W | – | – | √ | 0000H |
| FF15H | | | | | | | | |
| FF16H | 8 位定时器计数器 50 | | TM50 | R | – | √ | – | 00H |
| FF17H | 8 位定时器比较寄存器 50 | | CR50 | R/W | – | √ | – | 00H |
| FF18H | 8 位定时器 H 比较寄存器 00 | | CMP00 | R/W | – | √ | – | 00H |
| FF19H | 8 位定时器 H 比较寄存器 10 | | CMP10 | R/W | – | √ | – | 00H |
| FF1AH | 8 位定时器 H 比较寄存器 01 | | CMP01 | R/W | – | √ | – | 00H |
| FF1BH | 8 位定时器 H 比较寄存器 11 | | CMP11 | R/W | – | √ | – | 00H |
| FF1FH | 8 位定时器计数器 51 | | TM51 | R | – | √ | – | 00H |
| FF20H | 端口模式寄存器 0 | | PM0 | R/W | √ | √ | – | FFH |
| FF21H | 端口模式寄存器 1 | | PM1 | R/W | √ | √ | – | FFH |
| FF22H | 端口模式寄存器 2 | | PM2 | R/W | √ | √ | – | FFH |
| FF23H | 端口模式寄存器 3 | | PM3 | R/W | √ | √ | – | FFH |
| FF24H | 端口模式寄存器 4 | | PM4 | R/W | √ | √ | – | FFH |
| FF26H | 端口模式寄存器 6 | | PM6 | R/W | √ | √ | – | FFH |
| FF27H | 端口模式寄存器 7 | | PM7 | R/W | √ | √ | – | FFH |
| FF28H | A/D 转换器模式寄存器 | | ADM | R/W | √ | √ | – | 00H |
| FF29H | 模拟输入通道选择寄存器 | | ADS | R/W | √ | √ | – | 00H |
| FF2CH | 端口模式寄存器 12 | | PM12 | R/W | √ | √ | – | FFH |
| FF2EH | 端口模式寄存器 14[注] | | PM14 | R/W | √ | √ | – | FFH |
| FF2FH | A/D 端口配置寄存器 | | ADPC | R/W | √ | √ | – | 00H |

注　　仅 **48** 引脚产品可用。

**表 3-7. 特殊功能寄存器列表 (2/4)**

| 地址 | 特殊功能寄存器 (SFR) 名称 | 符号 | R/W | 可操作位单元 | | | 复位后 |
|---|---|---|---|---|---|---|---|
| | | | | 1 位 | 8 位 | 16 位 | |
| FF30H | 上拉电阻选择寄存器 0 | PU0 | R/W | √ | √ | – | 00H |
| FF31H | 上拉电阻选择寄存器 1 | PU1 | R/W | √ | √ | – | 00H |
| FF33H | 上拉电阻选择寄存器 3 | PU3 | R/W | √ | √ | – | 00H |
| FF34H | 上拉电阻选择寄存器 4 | PU4 | R/W | √ | √ | – | 00H |
| FF37H | 上拉电阻选择寄存器 7 | PU7 | R/W | √ | √ | – | 00H |
| FF3CH | 上拉电阻选择寄存器 12 | PU12 | R/W | √ | √ | – | 00H |
| FF3EH | 上拉电阻选择寄存器 14[注 1] | PU14 | R/W | √ | √ | – | 00H |
| FF40H | 时钟输出选择寄存器[注 1] | CKS | R/W | √ | √ | – | 00H |
| FF41H | 8 位定时器比较寄存器 51 | CR51 | R/W | – | √ | – | 00H |
| FF43H | 8 位定时器模式控制寄存器 51 | TMC51 | R/W | √ | √ | – | 00H |
| FF48H | 外部中断上升沿允许寄存器 | EGP | R/W | √ | √ | – | 00H |
| FF49H | 外部中断下降沿允许寄存器 | EGN | R/W | √ | √ | – | 00H |
| FF4FH | 输入切换控制寄存器 | ISC | R/W | √ | √ | – | 00H |
| FF50H | 异步串行接口操作模式寄存器 6 | ASIM6 | R/W | √ | √ | – | 01H |
| FF53H | 异步串行接口接收错误状态寄存器 6 | ASIS6 | R | – | √ | – | 00H |
| FF55H | 异步串行接口发送状态寄存器 6 | ASIF6 | R | – | √ | – | 00H |
| FF56H | 时钟选择寄存器 6 | CKSR6 | R/W | – | √ | – | 00H |
| FF57H | 波特率发生器控制寄存器 6 | BRGC6 | R/W | – | √ | – | FFH |
| FF58H | 异步串行接口控制寄存器 6 | ASICL6 | R/W | √ | √ | – | 16H |
| FF60H | 余数寄存器 0[注 2] | SDR0 SDR0L | R | – | √ | √ | 00H |
| FF61H | | SDR0H | | – | √ | | 00H |
| FF62H | 乘/除数寄存器A0[注 2] | MDA0L MDA0LL | R/W | – | √ | √ | 00H |
| FF63H | | MDA0LH | | – | √ | | 00H |
| FF64H | | MDA0H MDA0HL | R/W | – | √ | √ | 00H |
| FF65H | | MDA0HH | | – | √ | | 00H |
| FF66H | 乘/除数寄存器 B0[注 2] | MDB0 MDB0L | R/W | – | √ | √ | 00H |
| FF67H | | MDB0H | | – | √ | | 00H |
| FF68H | 乘法器/除法器控制寄存器 0[注 2] | DMUC0 | R/W | √ | √ | – | 00H |
| FF69H | 8 位定时器 H 模式寄存器 0 | TMHMD0 | R/W | √ | √ | – | 00H |
| FF6AH | 定时器时钟选择寄存器 50 | TCL50 | R/W | √ | √ | – | 00H |
| FF6BH | 8 位定时器模式控制寄存器 50 | TMC50 | R/W | √ | √ | – | 00H |
| FF6CH | 8 位定时器 H 模式寄存器 1 | TMHMD1 | R/W | √ | √ | – | 00H |
| FF6DH | 8 位定时器 H 载波控制寄存器 1 | TMCYC1 | R/W | √ | √ | – | 00H |
| FF6EH | 按键返回模式寄存器 | KRM | R/W | √ | √ | – | 00H |
| FF6FH | 钟表定时器操作模式寄存器 | WTM | R/W | √ | √ | – | 00H |
| FF70H | 异步串行接口操作模式寄存器 0 | ASIM0 | R/W | √ | √ | – | 01H |

注 **1.** 仅 48 引脚产品可用。
**2.** 仅 μPD78F0514, 78F0515, 和 78F0515D 可用。

表 3-7. 特殊功能寄存器列表 (3/4)

| 地址 | 特殊功能寄存器 (SFR) 名称 | 符号 | R/W | 可操作位单元 | | | 复位后 |
|---|---|---|---|---|---|---|---|
| | | | | 1 位 | 8 位 | 16 位 | |
| FF71H | 波特率发生器控制寄存器 0 | BRGC0 | R/W | – | √ | – | 1FH |
| FF72H | 接收缓冲寄存器 0 | RXB0 | R | – | √ | – | FFH |
| FF73H | 异步串行接口接收错误状态寄存器 0 | ASIS0 | R | – | √ | – | 00H |
| FF74H | 发送移位寄存器 0 | TXS0 | W | – | √ | – | FFH |
| FF80H | 串行操作模式寄存器 10 | CSIM10 | R/W | √ | √ | – | 00H |
| FF81H | 串行时钟选择寄存器 10 | CSIC10 | R/W | √ | √ | – | 00H |
| FF84H | 发送缓冲寄存器 10 | SOTB10 | R/W | – | √ | – | 00H |
| FF8CH | 定时器时钟选择寄存器 51 | TCL51 | R/W | √ | √ | – | 00H |
| FF99H | 看门狗定时器允许寄存器 | WDTE | R/W | – | √ | – | 1AH/9AH[注1] |
| FF9FH | 时钟操作模式选择寄存器 | OSCCTL | R/W | √ | √ | – | 00H |
| FFA0H | 内部振荡模式寄存器 | RCM | R/W | √ | √ | – | 80H[注2] |
| FFA1H | 主时钟模式寄存器 | MCM | R/W | √ | √ | – | 00H |
| FFA2H | 主 OSC 控制寄存器 | MOC | R/W | √ | √ | – | 80H |
| FFA3H | 振荡稳定时间计数器的状态寄存器 | OSTC | R | √ | √ | – | 00H |
| FFA4H | 振荡稳定时间选择寄存器 | OSTS | R/W | – | √ | – | 05H |
| FFA5H | IIC 移位寄存器 0 | IIC0 | R/W | – | √ | – | 00H |
| FFA6H | IIC 控制寄存器 0 | IICC0 | R/W | √ | √ | – | 00H |
| FFA7H | 从设备地址寄存器 0 | SVA0 | R/W | – | √ | – | 00H |
| FFA8H | IIC 时钟选择寄存器 0 | IICCL0 | R/W | √ | √ | – | 00H |
| FFA9H | IIC 功能扩展寄存器 0 | IICX0 | R/W | √ | √ | – | 00H |
| FFAAH | IIC 状态寄存器 0 | IICS0 | R | √ | √ | – | 00H |
| FFABH | IIC 标志寄存器 0 | IICF0 | R/W | √ | √ | – | 00H |
| FFACH | 复位控制标志寄存器 | RESF | R | – | √ | – | 00H[注3] |
| FFBAH | 16 位定时器模式控制寄存器 00 | TMC00 | R/W | √ | √ | – | 00H |
| FFBBH | 预分频模式寄存器 00 | PRM00 | R/W | √ | √ | – | 00H |
| FFBCH | 捕捉/比较控制寄存器 00 | CRC00 | R/W | √ | √ | – | 00H |
| FFBDH | 16 位定时器输出控制寄存器 00 | TOC00 | R/W | √ | √ | – | 00H |
| FFBEH | 低电压检测寄存器 | LVIM | R/W | √ | √ | – | 00H[注3] |
| FFBFH | 低电压检测等级选择寄存器 | LVIS | R/W | √ | √ | – | 00H[注3] |
| FFE0H | 中断请求标志寄存器 0L | IF0 IF0L | R/W | √ | √ | √ | 00H |
| FFE1H | 中断请求标志寄存器 0H | IF0H | R/W | √ | √ | | 00H |
| FFE2H | 中断请求标志寄存器 1L | IF1 IF1L | R/W | √ | √ | √ | 00H |
| FFE3H | 中断请求标志寄存器 1H | IF1H | R/W | √ | √ | | 00H |
| FFE4H | 中断屏蔽标志寄存器 0L | MK0 MK0L | R/W | √ | √ | √ | FFH |
| FFE5H | 中断屏蔽标志寄存器 0H | MK0H | R/W | √ | √ | | FFH |

注 **1.** WDTE 的复位值由选项字节的设置决定。

**2.** 复位后该寄存器的值立即变为 00H，但在等待高速内部振荡器的振荡精确稳定后，该寄存器值会自动变为 80H。

**3.** RESF，LVIM 和 LVIS 的复位值会根据复位源的变化而变化。

表 3-7. 特殊功能寄存器列表 (4/4)

| 地址 | 特殊功能寄存器 (SFR) 名称 | 符号 | | R/W | 可操作位单元 | | | 复位后 |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 位 | 8 位 | 16 位 | |
| FFE6H | 中断屏蔽标志寄存器 1L | MK1 | MK1L | R/W | √ | √ | √ | FFH |
| FFE7H | 中断屏蔽标志寄存器 1H | | MK1H | R/W | √ | √ | | FFH |
| FFE8H | 优先级指定标志寄存器 0L | PR0 | PR0L | R/W | √ | √ | √ | FFH |
| FFE9H | 优先级指定标志寄存器 0H | | PR0H | R/W | √ | √ | | FFH |
| FFEAH | 优先级指定标志寄存器 1L | PR1 | PR1L | R/W | √ | √ | √ | FFH |
| FFEBH | 优先级指定标志寄存器 1H | | PR1H | R/W | √ | √ | | FFH |
| FFF0H | 内部存储器容量切换寄存器[注] | IMS | | R/W | – | √ | – | CFH |
| FFF4H | 内部扩展RAM容量切换寄存器[注] | IXS | | R/W | – | √ | – | 0CH |
| FFFBH | 处理器时钟控制寄存器 | PCC | | R/W | √ | √ | – | 01H |

注 不管内存容量有多大，所有 78K0/KF2 产品的内存容量切换寄存器(IMS)和内部扩展 RAM 容量切换寄存器(IXS)的初始值都是固定的(IMS = CFH, IXS = 0CH)。因此每种产品设置值如下所示。

| Flash 存储器版本 (78K0/KC2) | IMS | IXS | ROM 容量 | 内部高速 RAM 容量 | 内部扩展 RAM 容量 |
|---|---|---|---|---|---|
| μPD78F0511 | 04H | 0CH | 16 KB | 768 字节 | – |
| μPD78F0512 | C6H | | 24 KB | 1 KB | |
| μPD78F0513, 78F0513D | C8H | | 32 KB | | |
| μPD78F0514 | CCH | 0AH | 48 KB | | 1 KB |
| μPD78F0515, 78F0515D | CFH | 08H | 60 KB | | 2 KB |

## 3.3 指令地址寻址

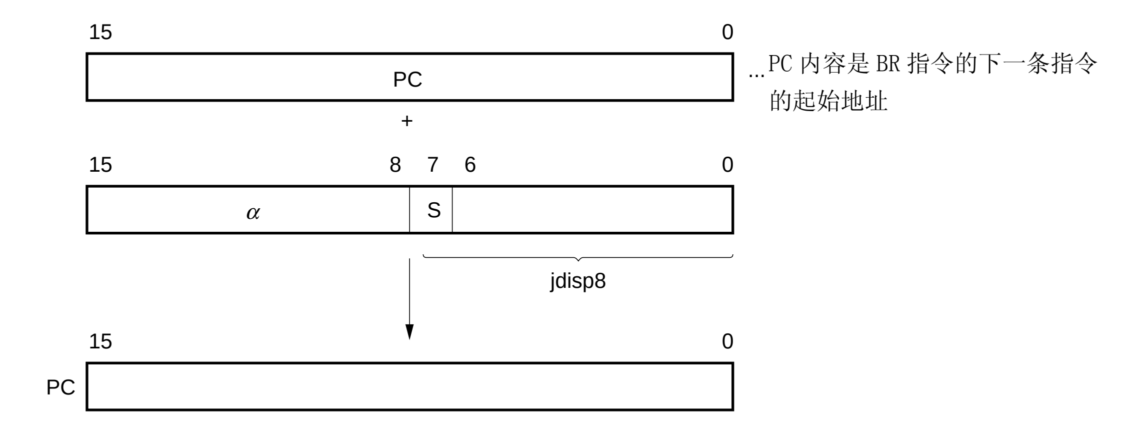
一条指令的地址是由程序计数器(PC)决定的。根据执行指令时所获取的下一条指令字节数，程序计数器(PC)的内容自动增加(每个字节加 1)。在执行转移指令时，将程序计数器(PC)的内容设置为转移目的地址，并按以下寻址方式确定地址。(要了解每条指令的详细信息，请参阅 **78K/0 系列指令用户手册(U12326E)**)。

### 3.3.1 相对寻址

**[功能]**

将一条指令的 8 位立即数(偏移量: jdisp8)与下一条指令的起始地址相加，结果赋给程序计数器(PC)，然后转向相加结果指向的地址。这个偏移量是带符号数的补码(–128 ～ +127)，其中第 7 位是符号位。
换句话说，在相对寻址中，分支的范围是从下一条指令起始地址的-128 到+127 之间。
当执行“BR $addr16”指令或条件转移指令时，将执行相对寻址功能。

**[图示]**




当S = 0， α 的所有位均为0
当 S = 1， α 的所有位均为 1

### 3.3.2 立即寻址

**[功能]**

将指令中的立即数赋给程序计数器(PC)，然后转向该地址。

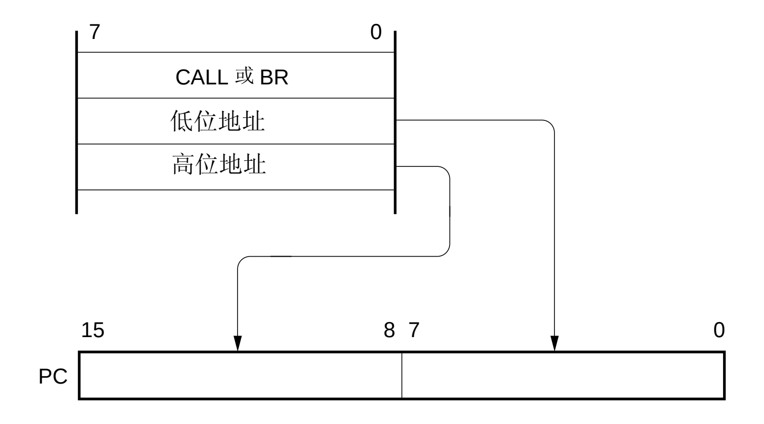
在执行"CALL !addr16"指令、 "BR !addr16"指令或 "CALLF !addr11" 指令时，将执行立即寻址功能。

CALL !addr16 和 BR !addr16 指令的转移地址范围是所有存储空间。

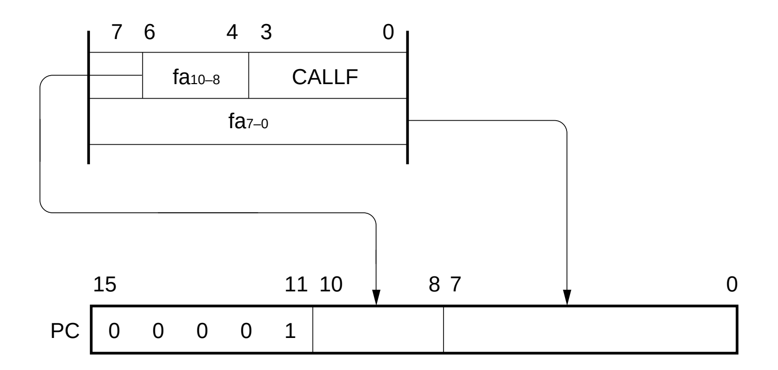
CALLF !addr11 指令的转移地址范围在 0800H 与 0FFFH 之间。

**[图示]**

CALL !addr16 和 BR !addr16 指令




CALLF !addr11 指令



### 3.3.3 表间接寻址

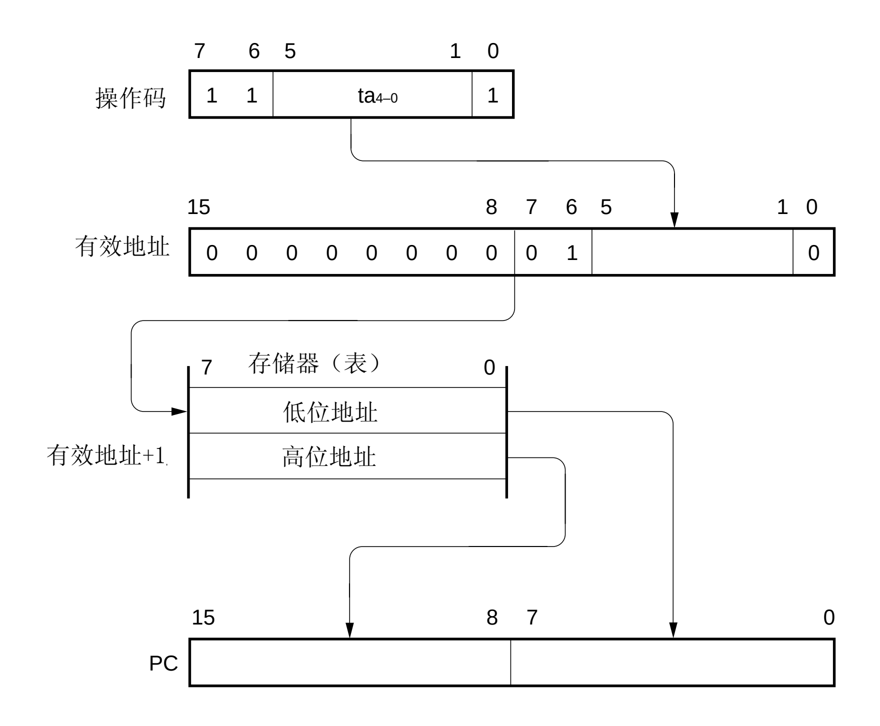
**[功能]**

通过指令码第 1 位到第 5 位的立即数，访问特定存储区中表的内容(转移目的地址)，并将表的内容赋给程序计数器(PC)，然后转向该地址执行程序。

在执行 CALLT [addr5]指令时，进行表间接寻址。

该指令访问的地址范围是表 40H～7FH 中所存储的地址，转移地址范围可以是整个存储器空间。

**[图示]**



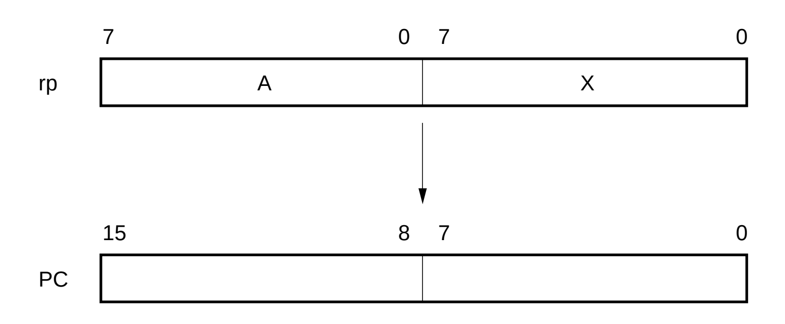

### 3.3.4 寄存器寻址

**[功能]**

将寄存器对(AX)的内容赋给程序计数器(PC)，然后转向该地址。

“BR AX”指令将执行寄存器寻址功能。

**[图示]**



## 3.4 操作数地址寻址

以下方法用来规定指令执行期间寄存器寻址和存储器寻址所进行的操作。

### 3.4.1 隐含寻址

**[功能]**

这种寻址方式自动寻址通用寄存器中作为累加器(A 和 AX)使用的寄存器。
在 78K0/KC2 系列指令中下列指令采用隐含寻址方式。

| 指令 | 隐含寻址所指定的寄存器 |
|---|---|
| MULU | A 寄存器存放被乘数，AX 寄存器存放运算结果 |
| DIVUW | AX 寄存器用于存放被除数和商 |
| ADJBA/ADJBS | 存放进行十进制调整后的数据 |
| ROR4/ROL4 | 存放用于数字循环的数字数据 |

**[操作数格式]**

由于指令自动采用隐含寻址方式，所以没有特定的操作数格式。

**[举例]**

以 MULU X 指令为例，这是一条 8 位乘 8 位的乘法运算指令，A 寄存器与 X 寄存器相乘的结果存放在 AX 中。在这个例子中 A 寄存器与 AX 寄存器均由隐含寻址方式指定。

### 3.4.2 寄存器寻址

**[功能]**

寄存器寻址方式将通用寄存器作为操作数进行访问，并由寄存器组选择标志(RBS0 ～ RBS1)和指令中的寄存器标识码，来指定需要访问的通用寄存器。

当具有下列操作数格式的指令执行时，采用寄存器寻址方式。如果使用 8 位寄存器，则指令码中有 3 位用来表示一个 8 位寄存器。

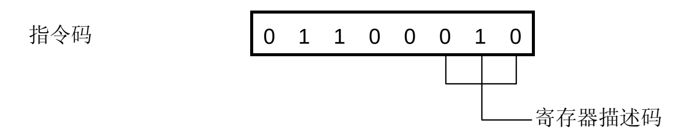
**[操作数格式]**

| 标识符 | 描述 |
|---|---|
| r | X, A, C, B, E, D, L, H |
| rp | AX, BC, DE, HL |

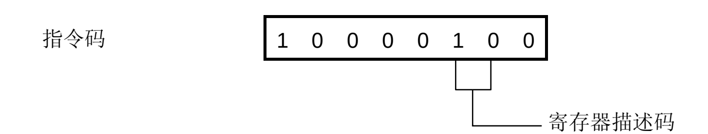
'r' 和 'rp'可用绝对名称(R0 ～ R7 以及 RP0 ～ RP3)和功能名称(X, A, C, B, E, D, L, H, AX, BC, DE 以及 HL)来描述。

**[举例]**

MOV A, C;　当选择通用寄存器 C 为“r”时




INCW DE;　当选择通用寄存器组 DE 为“rp”时



### 3.4.3 直接寻址

**[功能]**

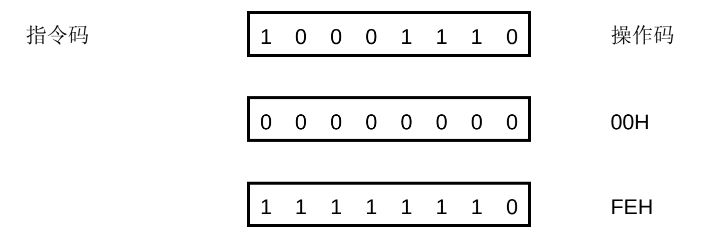
存储器会根据指令字中的操作数地址进行直接寻址操作。

**[操作数格式]**

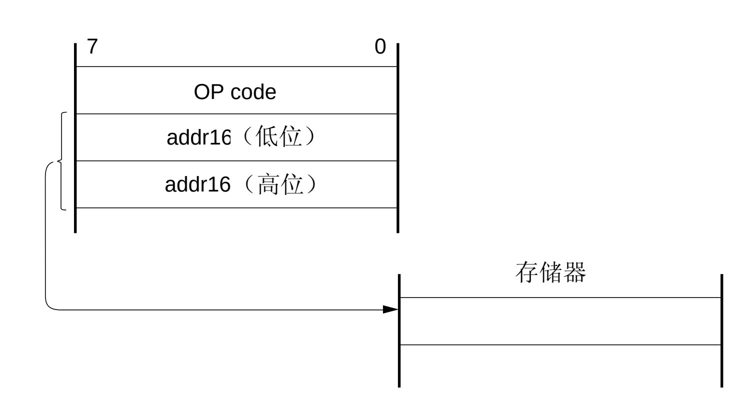
| 标识符 | 描述 |
| --- | --- |
| addr16 | 标号或 16 位立即数 |

**[举例]**

MOV A, !0FE00H; 将!addr16 设置为 FE00H 时




**[图示]**



### 3.4.4 短直接寻址

**[功能]**

用指令中 8 位立即数直接对存储器的固定操作区域寻址。

该方式的寻址范围是 FE20H～FF1FH 总共 256 字节的区域。内部 RAM 和特殊功能寄存器(SFR)分别映射在 FE20H ～ FEFFH 以及 FF00H ～ FF1FH 的区域。

采用短直接寻址方式的特殊功能寄存器(SFR)区域(FF00H ～ FF1FH)是整个特殊功能寄存器 SFR 区域的一部分。程序中经常访问的端口、用作定时器和事件计数器的比较和捕捉寄存器都被映射到该区域。这些特殊功能寄存器(SFR)可以用很少的字节数和时钟数进行操作。

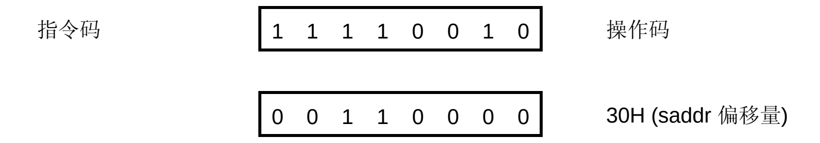
如果 8 位立即数是在 20H 和 FFH 之间，则将一个有效地址的第 8 位设置为 0；如果 8 位立即数是在 00H 与 1FH 之间，则一个有效地址的第 8 位设置为 1。参见下面的 **[图示]** 。

**[操作数格式]**

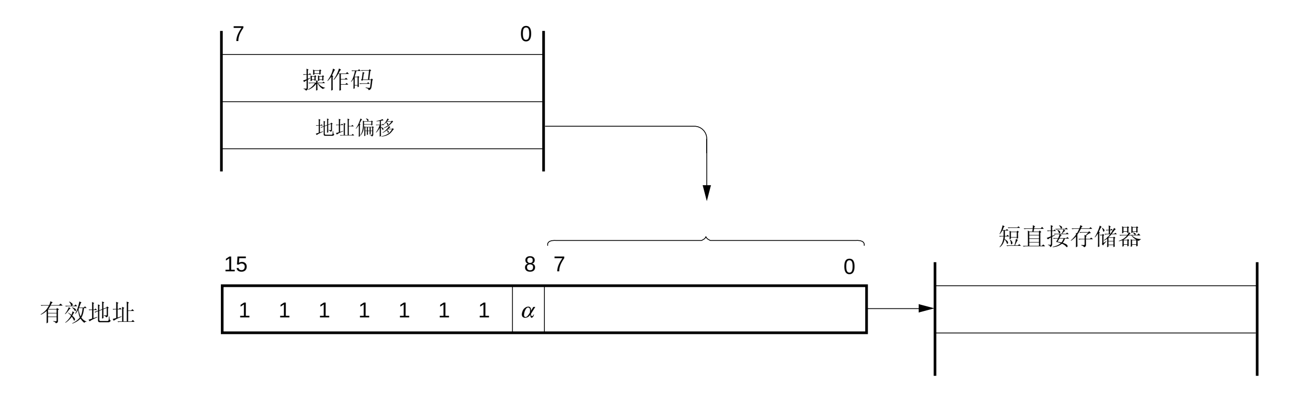
| 标识符 | 描述 |
|---|---|
| saddr | 标志或从 FE20H ～ FF1FH 的立即数 |
| saddrp | 标志或从 FE20H ～ FF1FH 的立即数 (仅使用偶地址) |

**[举例]**

```
LB1 EQU 0FE30H   ; 定义 FE30H 为 LB1
            :
MOV LB1, A       ; 当 LB1 代表 saddr 区域的 FE30H 地址并且 A 寄存器的值就传送给这个地址时。
```




**[图示]**




当8位立即数在20H与FFH之间时， α 等于0 。

当 8 位立即数的地址在 00H 与 1FH 之间时， α 等于 1。

### 3.4.5 特殊功能寄存器 (SFR) 寻址

**[功能]**

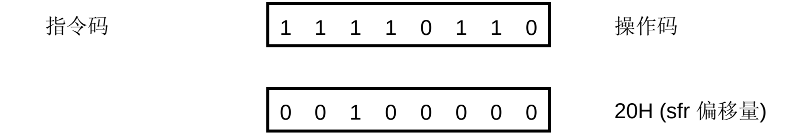
通过指令中的 8 位立即数对存储器的特殊功能寄存器(SFR)区域进行寻址。
寻址区间为 FF00H～FFCFH 以及 FFE0H～FFFFH，共 240 字节。而映射在 FF00H～FF1FH 区间的特殊功能寄存器则采用短直接寻址方式。

**[操作数格式]**

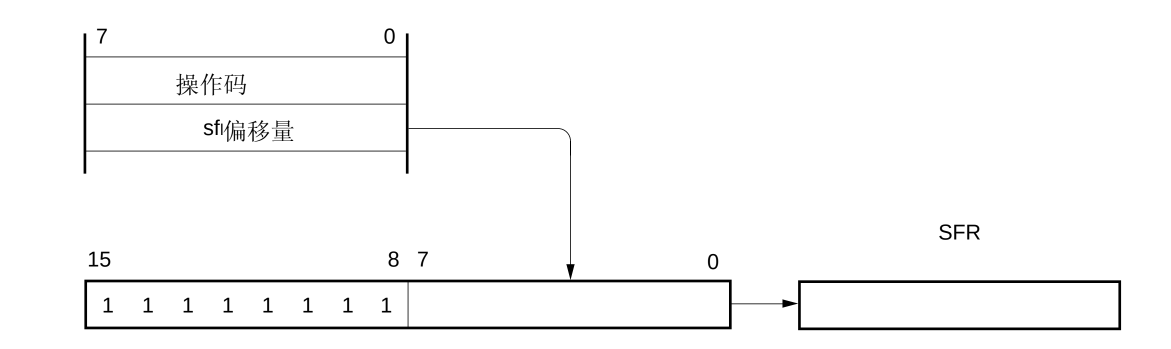
| 标识符 | 描述 |
|---|---|
| sfr | 特殊功能寄存器名 |
| sfrp | 16 位可操作特殊功能寄存器名 (仅使用偶地址) |

**[举例]**

MOV PM0, A; 选择 PM0(FF20H)作为 sfr




**[图示]**



### 3.4.6 寄存器间接寻址

**[功能]**

根据寄存器对的内容进行寻址。该寄存器对由寄存器组选择标志(RBS0 和 RBS1)和指令字中的寄存器对指定码指定。

**[操作数格式]**

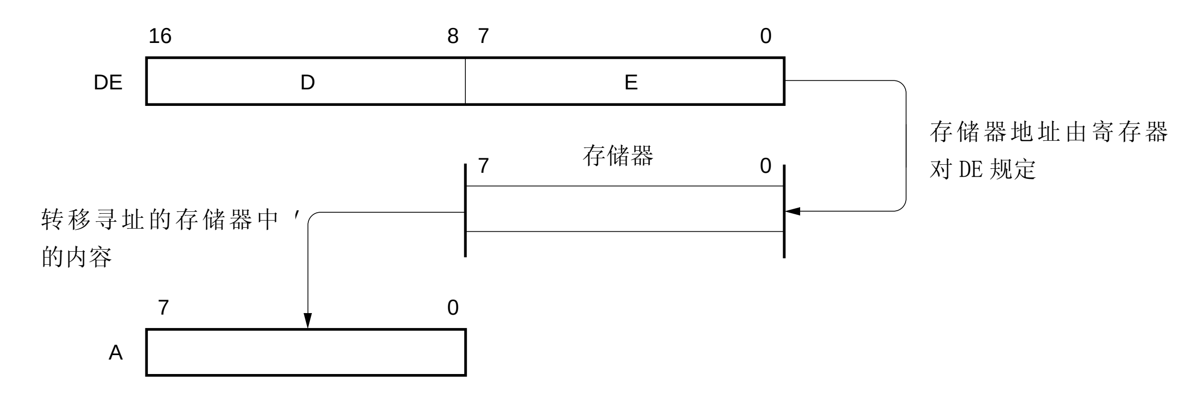
| 标识符 | 描述 |
|---|---|
| – | [DE], [HL] |

**[举例]**

MOV A, [DE]; 选择 DE 寄存器对作为操作数时

指令码 | 1 0 0 0 0 1 0 1 |

**[图示]**



### 3.4.7 基址寻址

**[功能]**

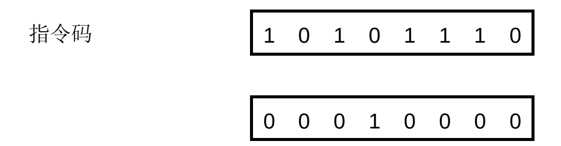
将 8 位立即数加到 HL 寄存器对中，HL 寄存器对作为基地址寄存器。根据相加结果寻址。需要访问的 HL 寄存器对属于由寄存器组选择标志(RBS0 和 RBS1)确定的寄存器组。通过将偏移量扩展为 16 位正数，来完成加法操作，第 16 位的进位忽略不计。

**[操作数格式]**

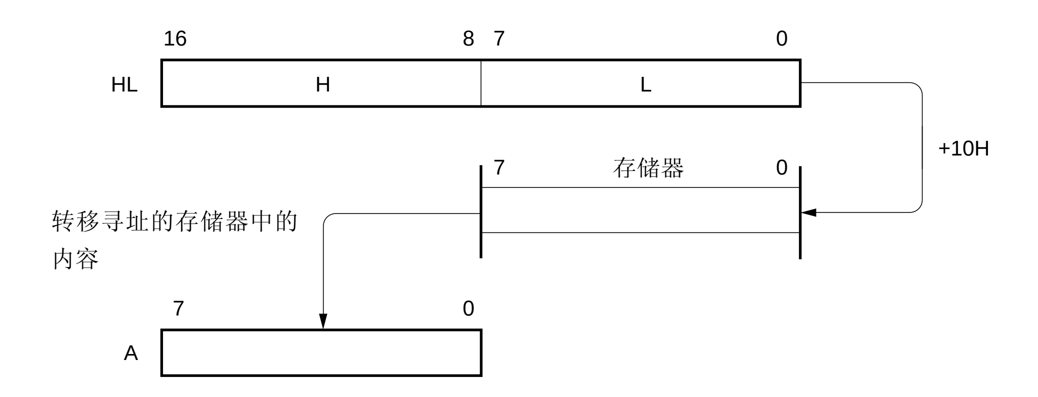
| 标识符 | 描述 |
|---|---|
| – | [HL + byte] |

**[举例]**

MOV A, [HL + 10H]; byte 的值为 10H 时




**[图示]**



### 3.4.8 基址变址寻址

**[功能]**

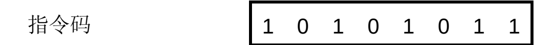
将 B 或 C 寄存器的内容加到 HL 寄存器中，HL 寄存器作为基地址寄存器，并根据相加结果去寻址。需要访问的 HL、B 和 C 寄存器属于由寄存器组选择标志(RBS0 和 RBS1)确定的寄存器组。通过将 B 或 C 寄存器扩展为一个 16 位的正数来完成加法运算，第 16 位的进位忽略不计。

**[操作数格式]**

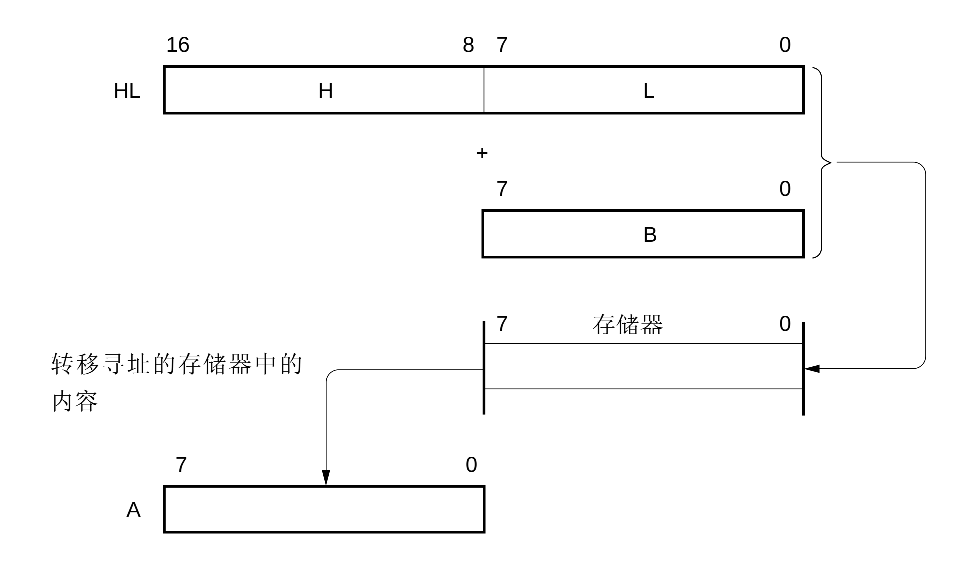
| 标识符 | 描述 |
|---|---|
| – | [HL + B], [HL + C] |

**[举例]**

MOV A, [HL +B]; 选择 B 寄存器




**[图示]**



### 3.4.9 堆栈寻址

**[功能]**

根据堆栈指针(SP)的内容对堆栈区域进行间接寻址。

当执行 PUSH, POP, 子程序调用和返回指令时，或者产生中断请求时保存或恢复寄存器操作时，将自动采用这种寻址方式。

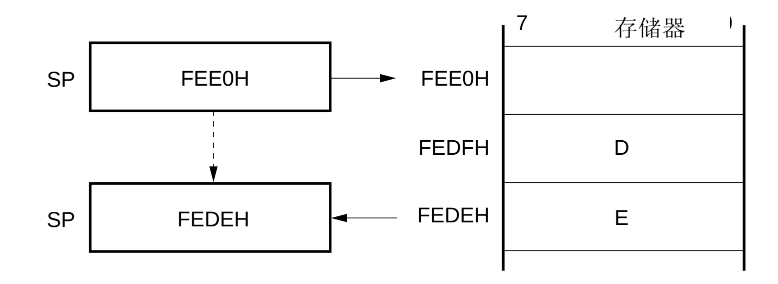
该方式仅对内部高速 RAM 区域进行寻址。

**[举例]**

PUSH DE; 保存在 DE 寄存器中

指令码 | 1 0 1 1 0 1 0 1 |

**[图示]**



## 4.1 端口功能

有两种类型的引脚 I/O 缓冲电源：$AV_{REF}$ 和 $V_{DD}$ 。这些电源和引脚之间的关系显示如下。

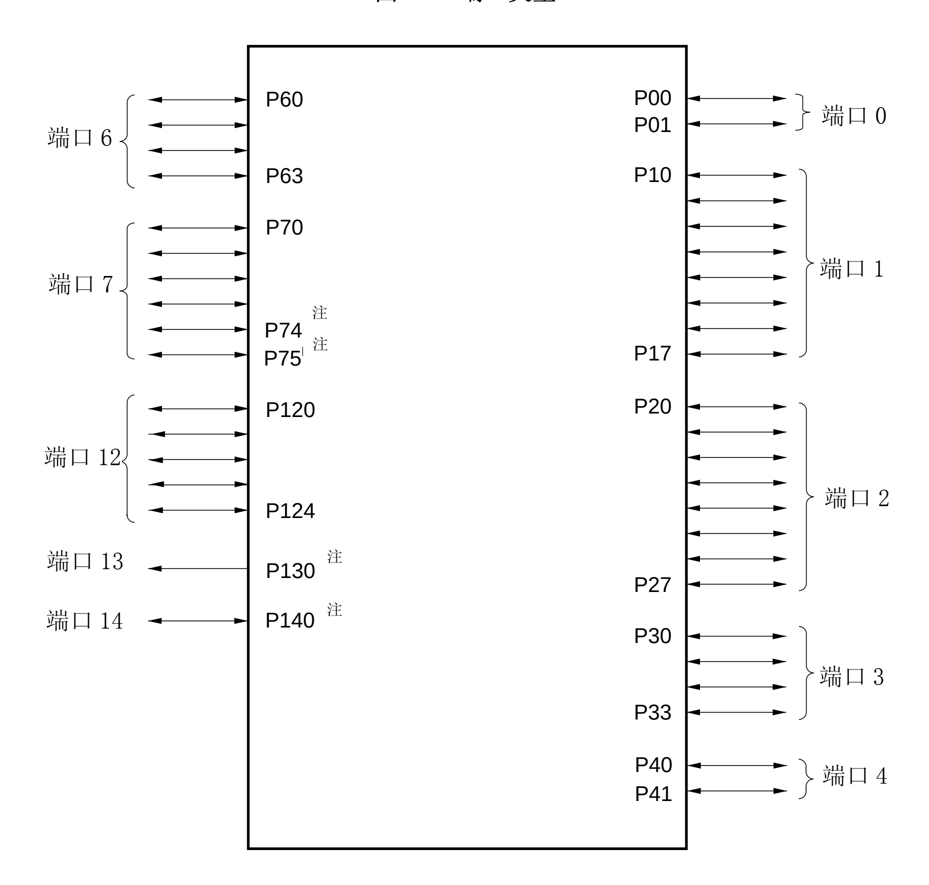
**表 4-1. 引脚 I/O 缓冲电源**

| 电源 | 相关引脚 |
| --- | --- |
| $AV_{REF}$ | P20 ～ P27 |
| $V_{DD}$ | 除 P20 ～ P27 的其它引脚 |

78K0/KC2 产品中提供的端口如 图 4-1 所示，这些端口可以支持多种控制操作。每个端口的功能如 表 4-2 所示。

除了作为数字 I/O 端口功能，这些端口还有几个复用功能。如需了解这些端口复用功能的详细信息，请参见**第二章引脚功能**。

**图 4-1. 端口类型**




**注** 仅 48 引脚产品。

表 4-2. 端口功能 (1/2)

| 功能名 | I/O | 功能 | 复位后 | 复用功能 |
| --- | --- | --- | --- | --- |
| P00 | I/O | 端口 0<br>2 位 I/O 端口<br>可以位选输入输出模式<br>通过软件设置，可以定义内部上拉电阻的使用 | 输入端口 | TI000 |
| P01 | | | | TI010/TO00 |
| P10 | I/O | 端口 1<br>8 位 I/O 端口<br>可以位选输入输出模式<br>通过软件设置，可以定义内部上拉电阻的使用 | 输入端口 | $\overline{SCK10}$/TxD0 |
| P11 | | | | SI10/RxD0 |
| P12 | | | | SO10 |
| P13 | | | | TxD6 |
| P14 | | | | RxD6 |
| P15 | | | | TOH0 |
| P16 | | | | TOH1/INTP5 |
| P17 | | | | TI50/TO50 |
| P20 ～ P27 | I/O | 端口 2<br>8 位 I/O 端口<br>可以位选输入输出模式 | 模拟输入 | ANI0 ～ ANI7 |
| P30 | I/O | 端口 3<br>4 位 I/O 端口<br>可以位选输入输出模式<br>通过软件设置，可以定义内部上拉电阻的使用 | 输入端口 | INTP1 |
| P31 | | | | INTP2/OCD1A[注 1] |
| P32 | | | | INTP3/OCD1B[注 1] |
| P33 | | | | TI51/TO51/INTP4 |
| P40 和 P41 | I/O | 端口 4<br>2 位 I/O 端口<br>可以位选输入输出模式<br>通过软件设置，可以定义内部上拉电阻的使用 | 输入端口 | – |
| P60 | I/O | 端口 6<br>4 位 I/O 端口<br>P60～P63 的输出是 N-ch 漏极开路输出(6 V 耐压)<br>可以位选输入输出模式 | 输入端口 | SCL0 |
| P61 | | | | SDA0 |
| P62 | | | | EXSCL0 |
| P63 | | | | – |
| P70 ～ P73 | I/O | 端口 7<br>6 位 I/O 端口<br>可以位选输入输出模式<br>通过软件设置，可以定义内部上拉电阻的使用 | 输入 | KR0 ～ KR3 |
| P74[注 2] 和 P75[注 2] | | | | – |
| P120 | I/O | 端口 12<br>5 位 I/O 端口<br>可以位选输入输出模式<br>只有 P120，可以通过软件设置，定义内部上拉电阻的使用 | 输入端口 | INTP0/EXLVI |
| P121 | | | | X1/OCD0A[注 1] |
| P122 | | | | X2/EXCLK/<br>OCD0B[注 1] |
| P123 | | | | XT1 |
| P124 | | | | XT2/EXCLKS |
| P130[注 2] | 输出 | 端口 13<br>1 位仅输出端口 | 输出端口 | – |

注 **1.** 仅限 μPD78F0513D 和 78F0515D 。
**2.** 仅限 48 引脚产品。

表 **4-2.** 端口功能 **(2/2)**

| 功能名 | I/O | 功能 | 复位后 | 复用功能 |
|---|---|---|---|---|
| P140[注] | I/O | 端口 14<br>1 位 I/O 端口<br>可以位选输入输出模式<br>通过软件设置，可以定义内部上拉电阻的使用 | 输入端口 | PCL/INTP6[注] |

**注**　仅限 48 引脚产品。

## 4.2 端口设置

端口包括如下硬件。

表 **4-3.** 端口配置

| 项目 | 配置 |
|---|---|
| 控制寄存器 | 端口模式寄存器 (PM0 ～ PM4, PM6, PM7, PM12, PM14[注])<br>端口寄存器 (P0 ～ P4, P6, P7, P12, P13[注], P14[注])<br>上拉电阻选择寄存器 (PU0, PU1, PU3, PU4, PU7, PU12, PU14[注])<br>A/D端口配置寄存器 (ADPC) |
| 端口 | • 44 引脚产品<br>总数: 37 (CMOS I/O: 32, CMOS 输出: 1, N-ch 漏极开路 I/O: 4)<br>• 48 引脚产品<br>总数: 41 (CMOS I/O: 36, CMOS 输出: 1, N-ch 漏极开路 I/O: 4) |
| 上拉电阻 | • 44 引脚产品　总数: 21<br>• 48 引脚产品　总数: 24 |

**注**　仅限 48 引脚产品。

### 4.2.1 端口 0

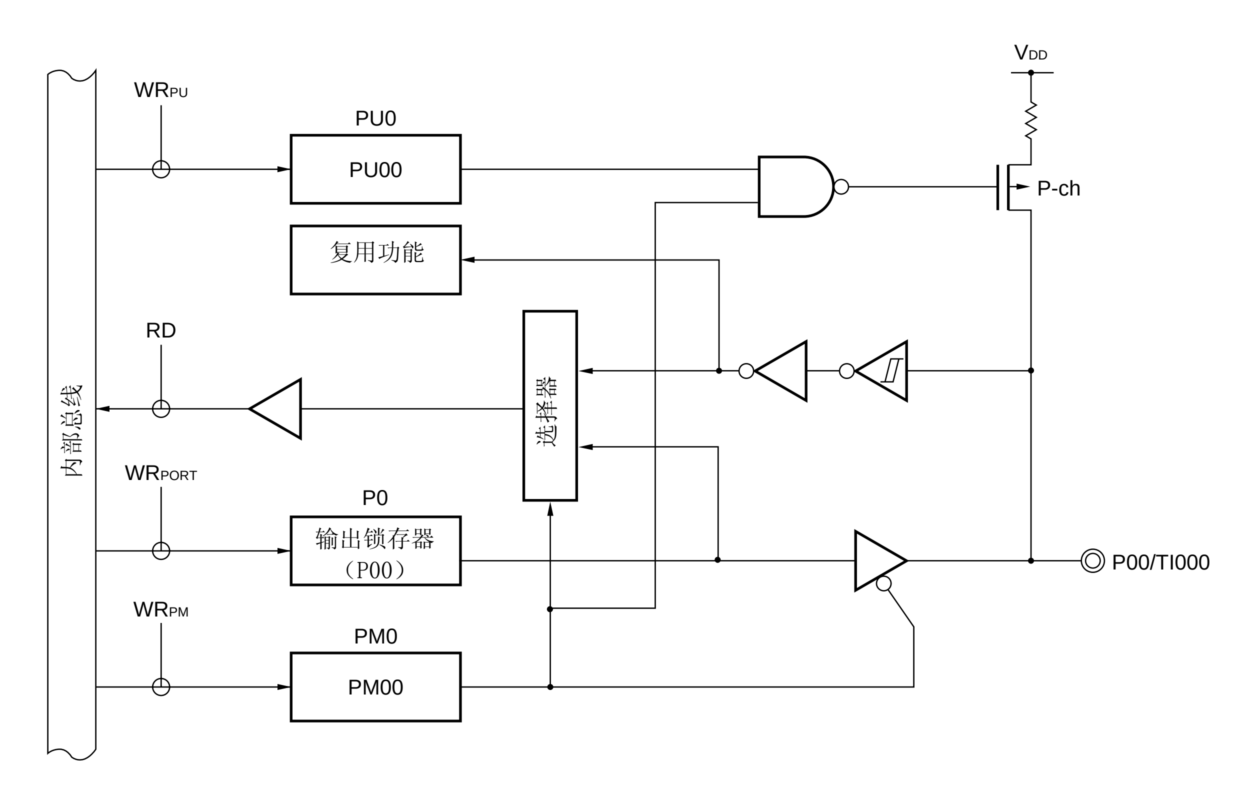
端口 0 是具有输出锁存功能的 2 位 I/O 端口。通过使用端口模式寄存器 0(PM0)，可以位选端口 0 为输入或输出模式。如果 P00 和 P01 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 0(PU0)以 1 位单元的方式指定。

这个端口还可以用于定时器 I/O 。

复位信号的产生可将端口 0 设置为输入模式。

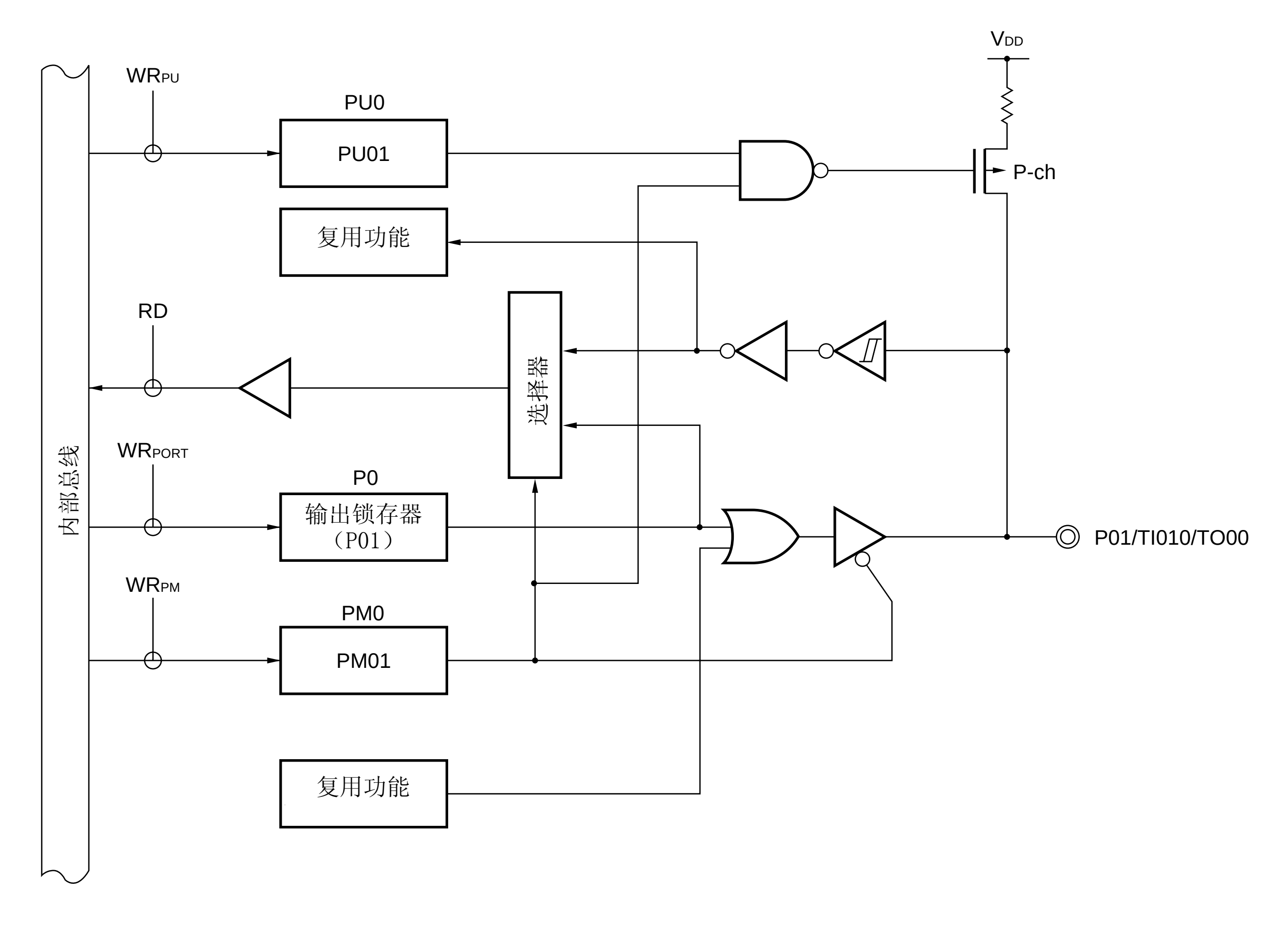
图 4-2 和 4-3 显示端口 0 的框图。

**图 4-2. P00 的框图**




P0: 端口寄存器 0
PU0: 上拉电阻选择寄存器 0
PM0: 端口模式寄存器 0
RD: 读信号
WR××: 写信号

图 **4-3. P01** 的框图




P0:　端口寄存器 0
PU0:　上拉电阻选择寄存器 0
PM0:　端口模式寄存器 0
RD:　读信号
WR××:　写信号

### 4.2.2 端口 1

端口 1 是具有输出锁存功能的 8 位 I/O 端口。通过使用端口模式寄存器 1(PM1)，可以位选端口 1 为输入或输出模式。如果 P10 ～ P17 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 1(PU1)以 1 位单元的方式指定。

这个端口还可以用于外部中断请求输入、串行接口数据 I/O、时钟 I/O 和定时器 I/O。

复位信号的产生可将端口 1 设置为输入模式。

图 4-4 ～ 4-8 显示端口 1 的框图。

**注意事项　如果 P10/$\overline{\text{SCK10}}$/TxD0 和 P12/SO10 作为通用端口使用，则将串行操作模式寄存器 10(CSIM10)和串行时钟选择寄存器 10(CSIC10)设置为默认状态（00H）。**

**图 4-4. P10 的框图**



P1:　端口寄存器 1

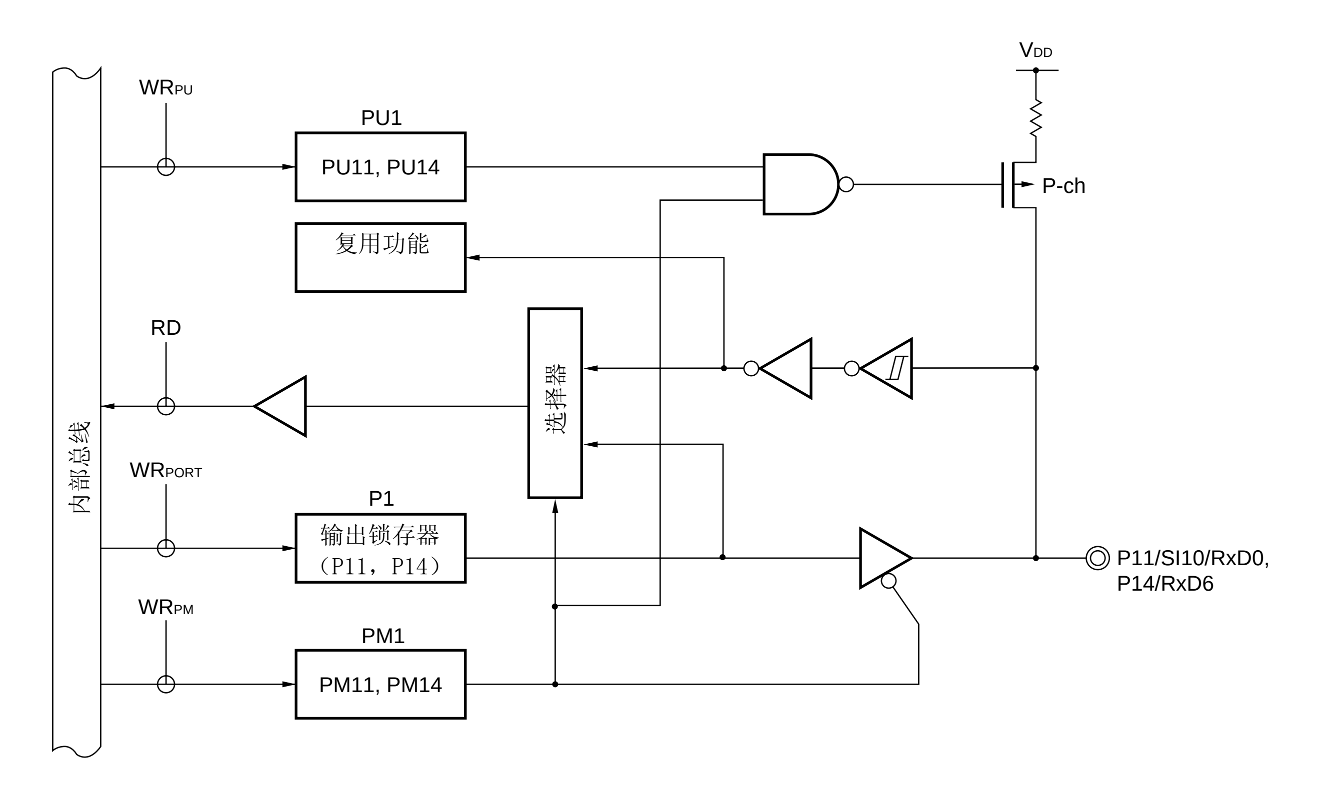
PU1:　上拉电阻选择寄存器 1

PM1:　端口模式寄存器 1

RD:　读信号

WR××:　写信号

**图 4-5. P11 和 P14 的框图**




P1:　　端口寄存器 1

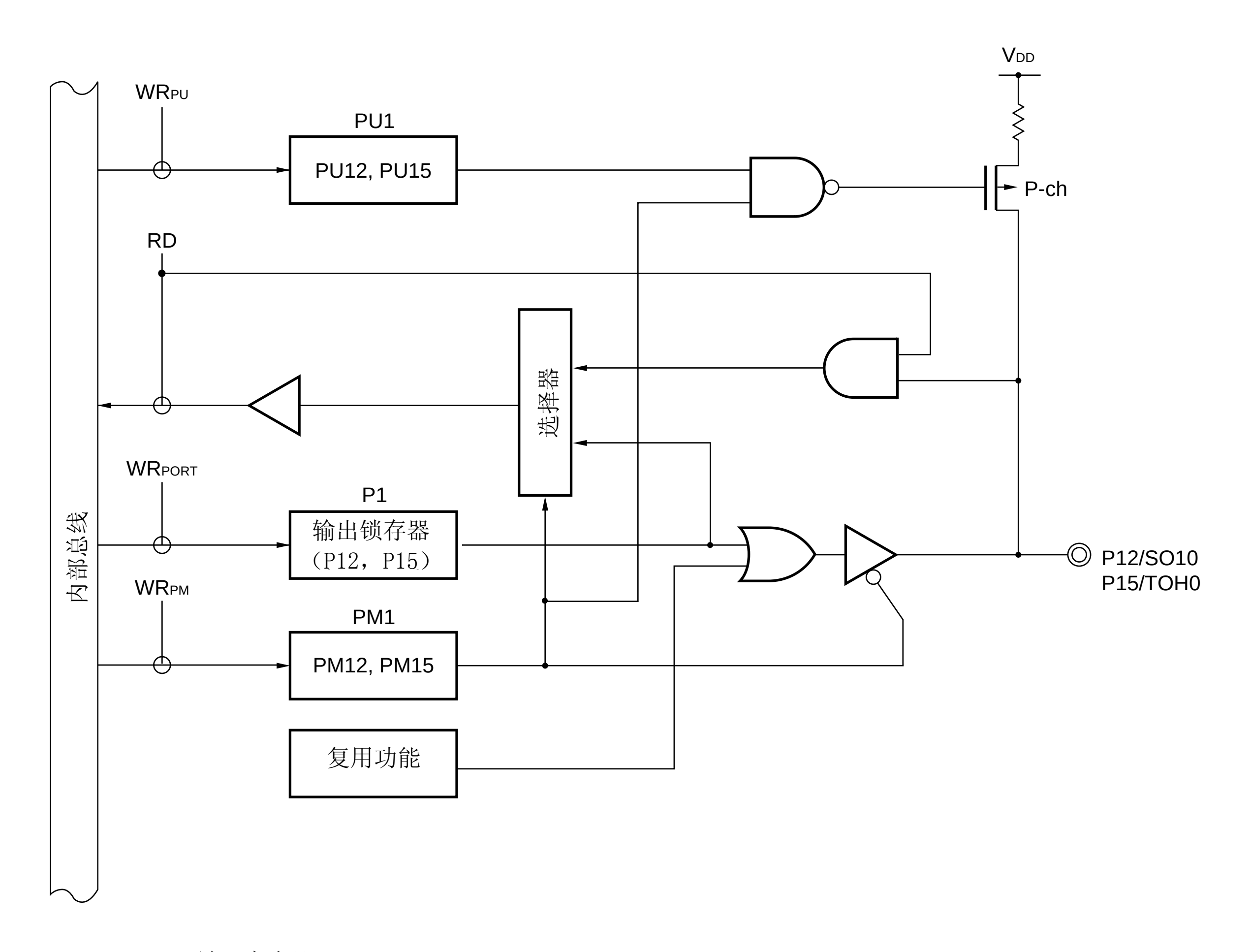
PU1:　　上拉电阻选择寄存器 1

PM1:　　端口模式寄存器 1

RD:　　读信号

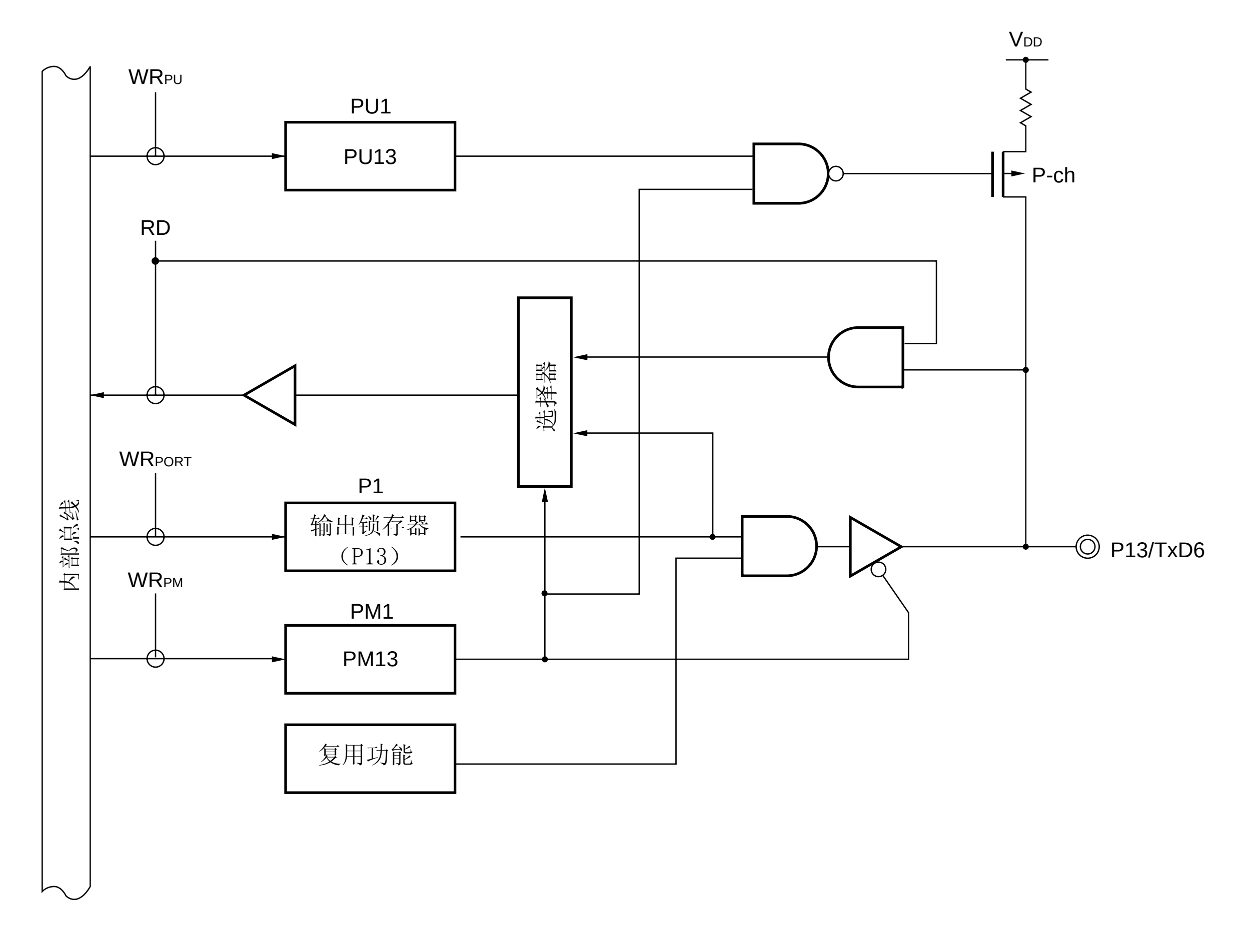
WR××:　写信号

图 **4-6. P12 和 P15** 的框图




P1:　端口寄存器 1
PU1:　上拉电阻选择寄存器 1
PM1:　端口模式寄存器 1
RD:　读信号
WR××:　写信号

图 **4-7. P13** 的框图




P1:　端口寄存器 1

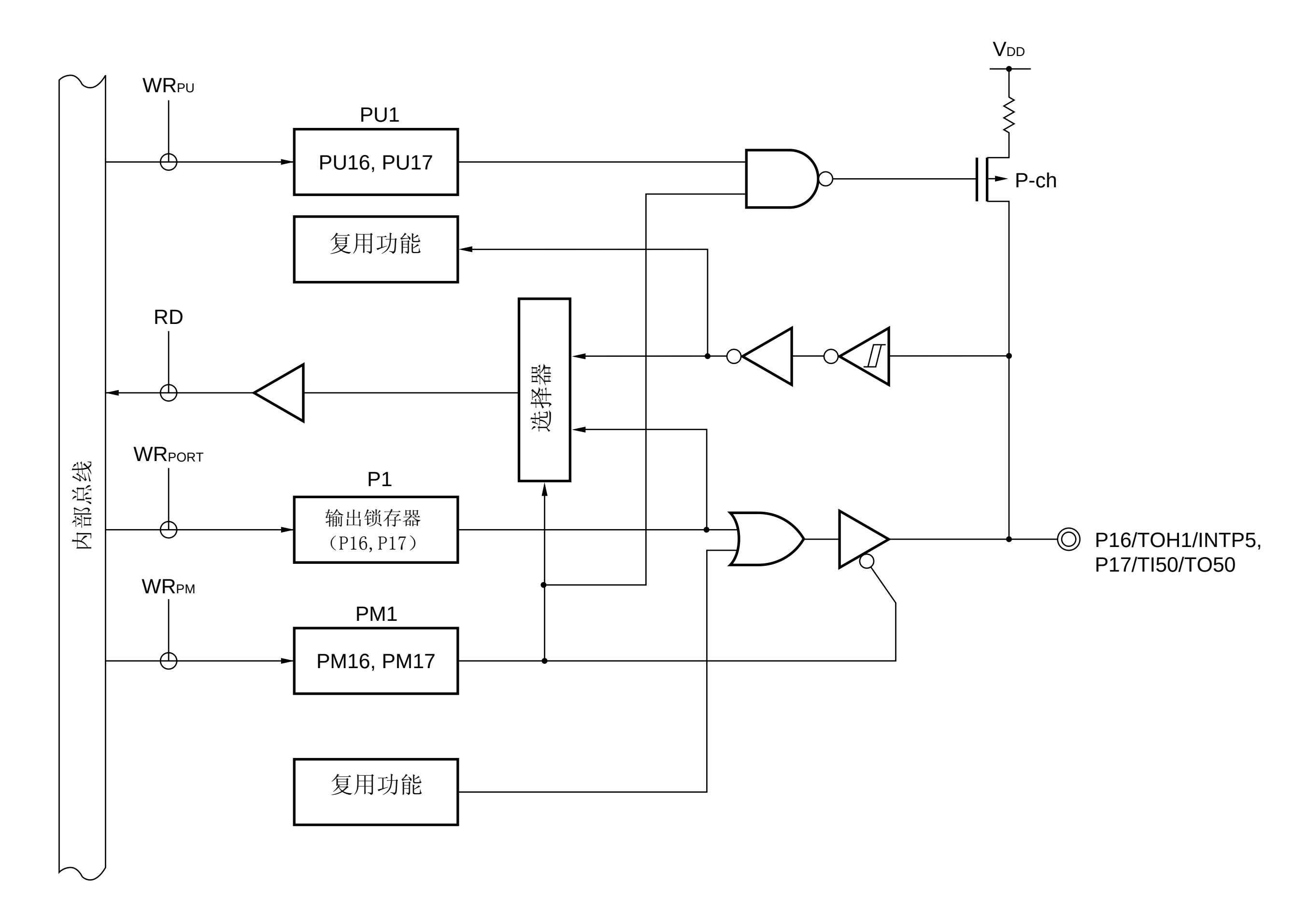
PU1:　上拉电阻选择寄存器 1

PM1:　端口模式寄存器 1

RD:　读信号

WR××:　写信号

图 **4-8. P16** 和 **P17** 的框图




P1:　　端口寄存器 1
PU1:　上拉电阻选择寄存器 1
PM1:　端口模式寄存器 1
RD:　　读信号
WR××:　写信号

### 4.2.3 端口 2

端口 2 是具有输出锁存功能的 8 位 I/O 端口。通过使用端口模式寄存器 2(PM2)，可以位选端口 2 为输入或输出模式。

这个端口还可以用于 A/D 转换器模拟输入。

如果要将 P20/ANI0 ～ P27/ANI7 作为数字输入引脚，应通过使用 A/D 端口配置寄存器（ADPC）将这些引脚设置为数字 I/O 模式，并通过使用 PM2 设置为输入模式。从低位开始使用这些引脚。

如果要将 P20/ANI0 ～ P27/ANI7 作为数字输出引脚，应通过使用 A/D 端口配置寄存器（ADPC）将这些引脚设置为数字 I/O 模式，并通过使用 PM2 设置为输出模式。

<R> **表 4-4. P20/ANI0 ～ P27/ANI7 引脚的功能设置**

| ADPC | PM2 | ADS | P20/ANI0 ～ P27/ANI7 引脚 |
|---|---|---|---|
| 数字 I/O 选择 | 输入模式 | – | 数字输入 |
| | 输出模式 | – | 数字输出 |
| 模拟输入选择 | 输入模式 | 选择 ANI. | 模拟输入 (被转换) |
| | | 不选择 ANI. | 模拟输入 (不被转换) |
| | 输出模式 | 选择 ANI. | 禁止设置 |
| | | 不选择 ANI. | |

在复位信号的产生时，P20/ANI0 ～ P27/ANI7 都被设置为模拟输入模式。

图 4-9 显示了端口 2 的框图。

<R> **注意事项　当端口 2 用于数字端口时，应使 $AV_{REF}$ 引脚与 $V_{DD}$ 引脚的电势相同。**

**图 4-9. P20 ～ P27 的框图**

P2:　端口寄存器 2

PM2:　端口模式寄存器 2

RD:　读信号

WR××:　写信号

### 4.2.4 端口 3

端口 3 是具有输出锁存功能的 4 位 I/O 端口。通过使用端口模式寄存器 3(PM3)，可以位选端口 3 为输入或输出模式。如果 P30 ～ P33 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 3(PU3)以 1 位单元的方式指定。

这个端口还可以用于外部中断请求输入和定时器 I/O。

复位信号的产生可将端口 3 设置为输入模式。

图 4-10 和 4-11 显示端口 3 的框图。

注意事项 **1. 在使用具有片上调试功能的产品 (μPD78F0513D 和 78F0515D) 时，应确保在复位以前将引脚 P31/INTP2/OCD1A**[注] **下拉，防止故障发生。**

<R> **2. 对于具有 48KB或更大Flash存储器而没有片上调试功能的产品 (μPD78F0514 和 78F0515)，有一个 “I”, “K”, 或 “E” 的产品等级，对于具有片上调试功能的产品 (μPD78F0513D 和 78F0515D) 在使用 Flash编程器写Flash存储器时，应按以下连接P31/INTP2/OCD1A**[注]**。**

**• P31/INTP2/OCD1A**[注]**: 通过电阻连接到 $V_{SS}$ (10 kΩ: 推荐)。**

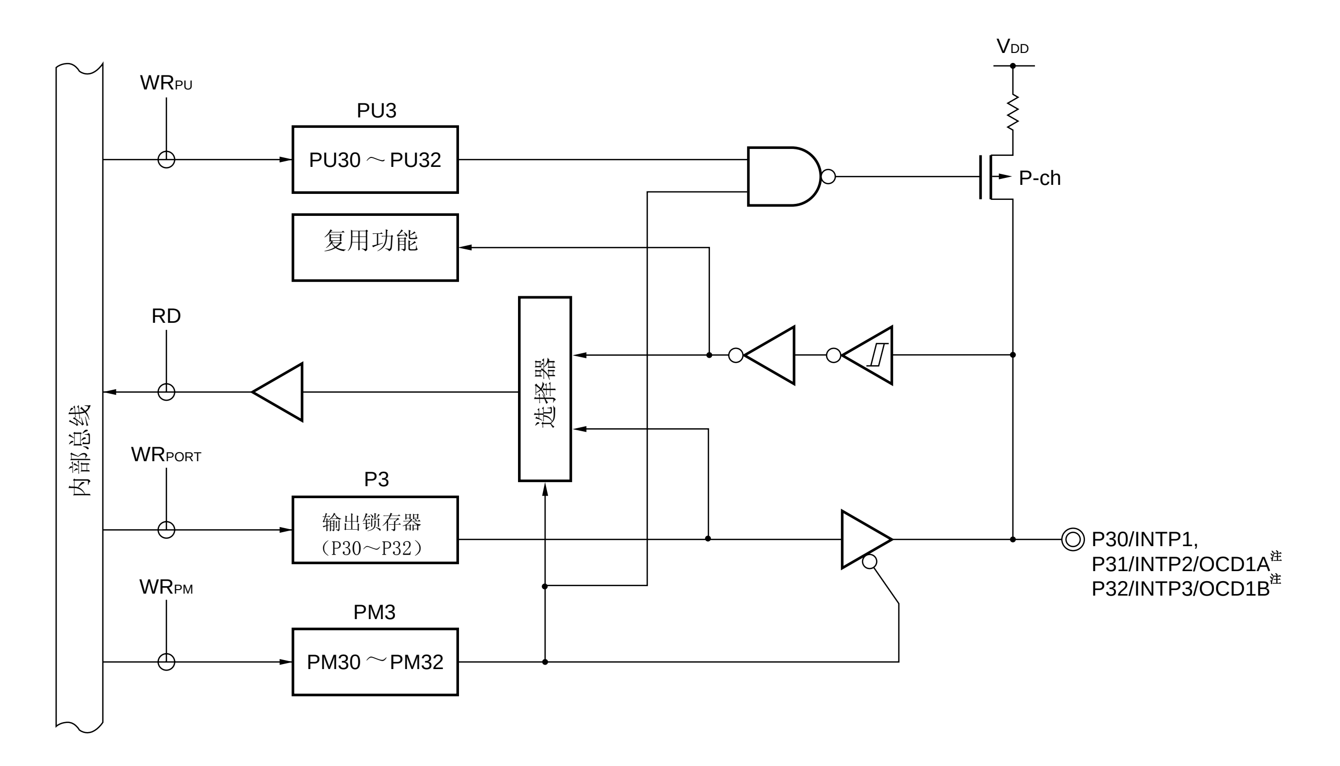
**当通过自编程方式写入 Flash 存储器时，以上连接并不需要。**

<R> 注　只有μPD78F0513D 和 78F0515D 提供 OCD1A 。

<R> 备注 **1.** 对于产品级别，请咨询 NEC 电子的销售代表。

**2.** 只有μPD78F0513D 和 78F0515D，在使用片上调试功能时，P31 和 P32 能够用于片上调试模式设定引脚 (OCD1A, OCD1B) 。对于怎样连接支持片上调试功能的在线仿真器 (QB-78K0MINI) ，请参看**第 26 章 片上调试功能 (仅μPD78F0513D 和 78F0515D)** 。

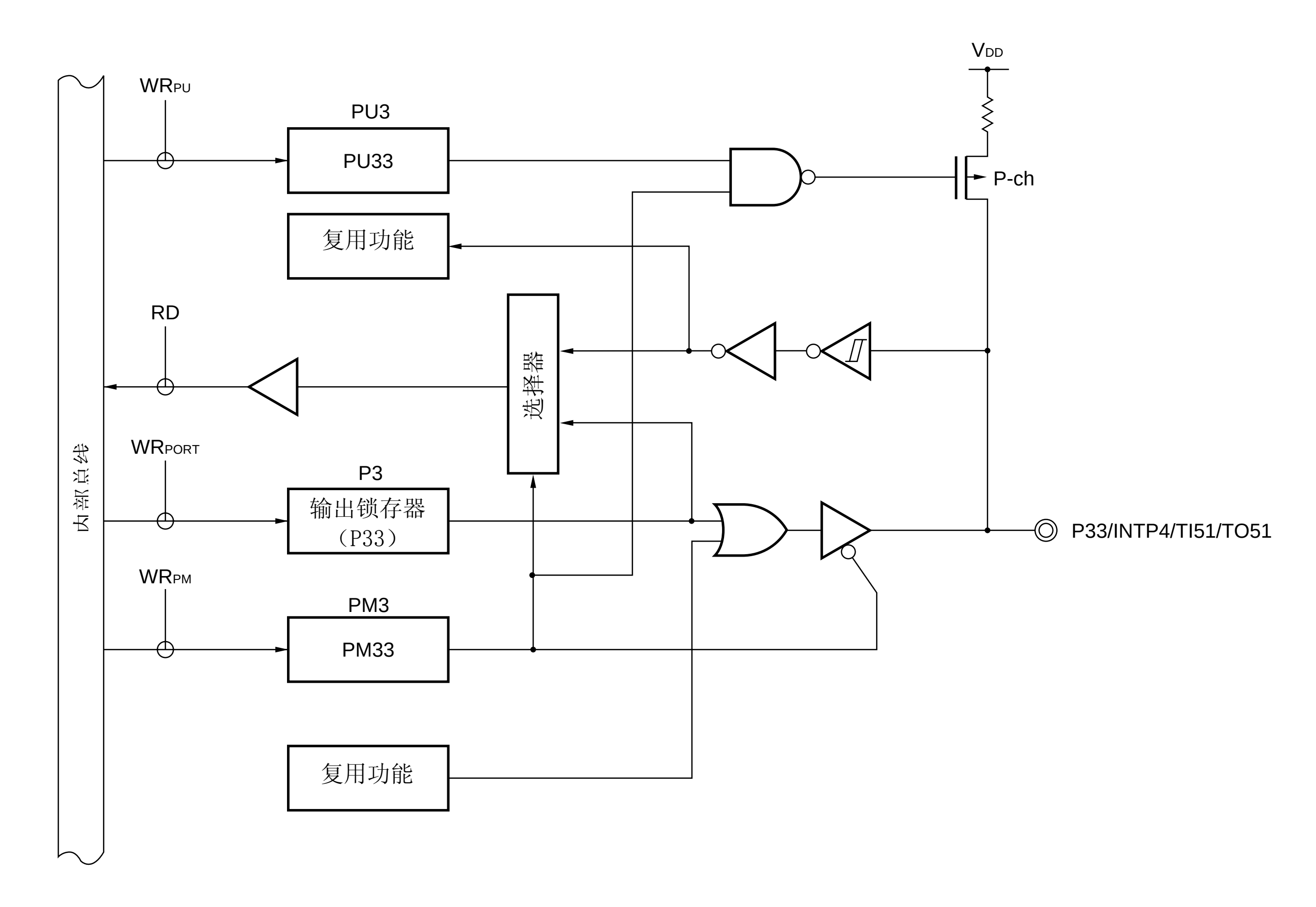
图 **4-10. P30 ～ P32** 的框图




P3:　　端口寄存器 3
PU3:　　上拉电阻选择寄存器 3
PM3:　　端口模式寄存器 3
RD:　　读信号
WR××:　写信号

注　　仅限 $\mu$PD78F0513D 和 78F0515D 。

**图 4-11. P33 的框图**




P3:　　端口寄存器 3
PU3:　上拉电阻选择寄存器 3
PM3:　端口模式寄存器 3
RD:　　读信号
WR××:　写信号

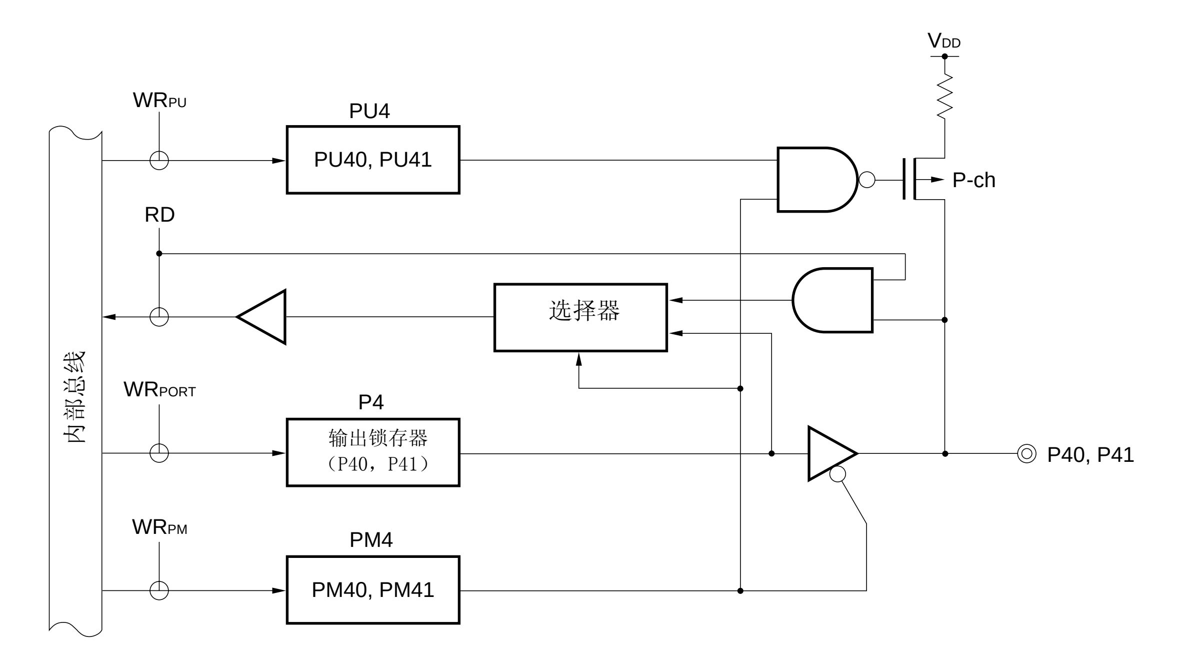
### 4.2.5 端口 4

端口 4 是具有输出锁存功能的 2 位 I/O 端口。通过使用端口模式寄存器 4(PM4)，可以位选端口 4 为输入或输出模式。如果 P40 和 P41 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 4(PU4)以 1 位单元的方式指定。

复位信号的产生可将端口 4 设置为输入模式。

图 4-12 显示端口 4 的框图。

**图 4-12. P40, P41 的框图**




P4:　　端口寄存器 4

PU4:　上拉电阻选择寄存器 4

PM4:　端口模式寄存器 4

RD:　　读信号

WR××:　写信号

### 4.2.6 端口 6

端口 6 是具有输出锁存功能的 4 位 I/O 端口。通过使用端口模式寄存器 6(PM6)，可以位选端口 6 为输入或输出模式。

P60 ～ P63 引脚的输出是 N-ch 漏极开路输出（6V 耐压）。

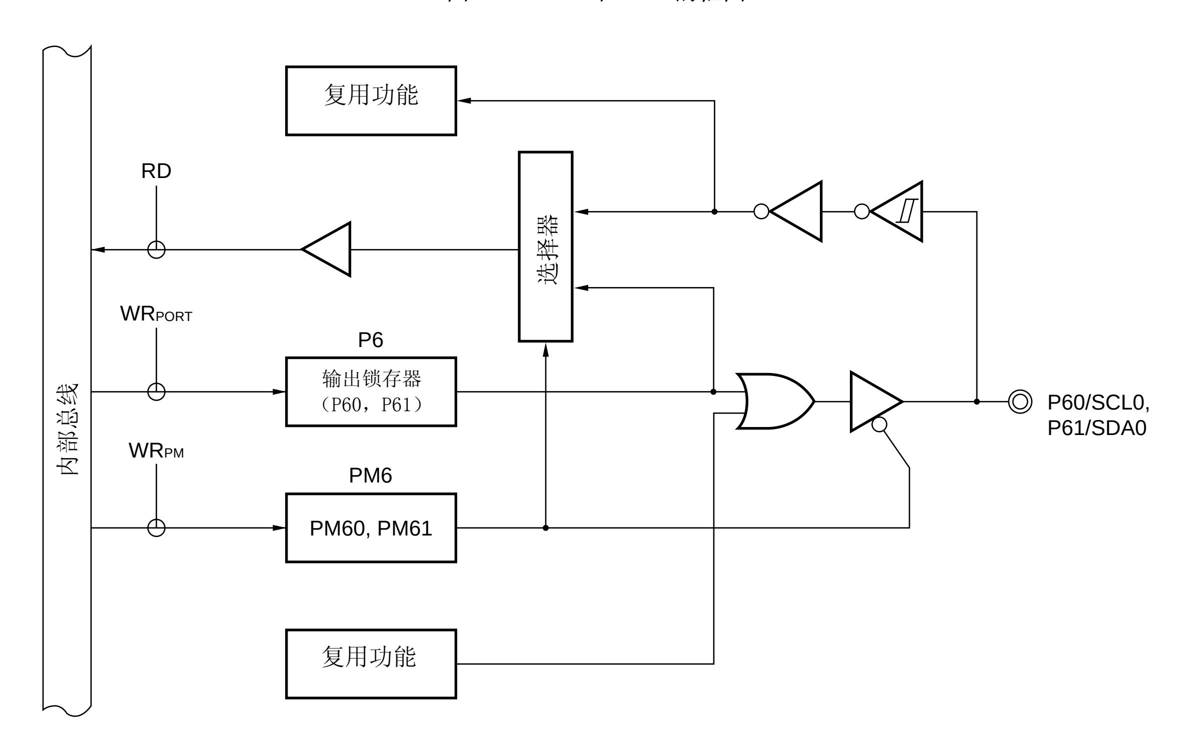
这个端口还可以用于串行接口数据 I/O、时钟 I/O 和外部时钟输入。

复位信号的产生可将端口 6 设置为输入模式。

图 4-13 ～ 4-15 显示端口 6 的框图。

**备注**　　在使用 P62/EXSCL0 作为串行接口外部时钟输入引脚时，应输入一个 6.4 MHz 的时钟。

**图 4-13. P60 和 P61 的框图**



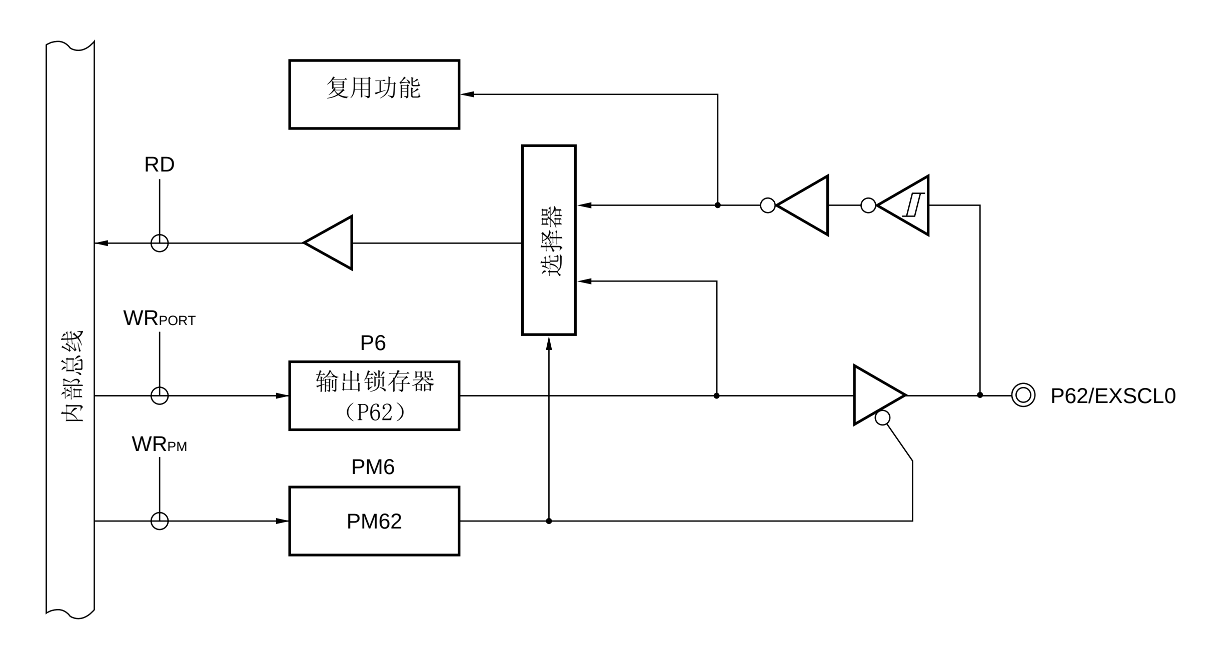

P6:　端口寄存器 6

PM6:　端口模式寄存器 6

RD:　读信号

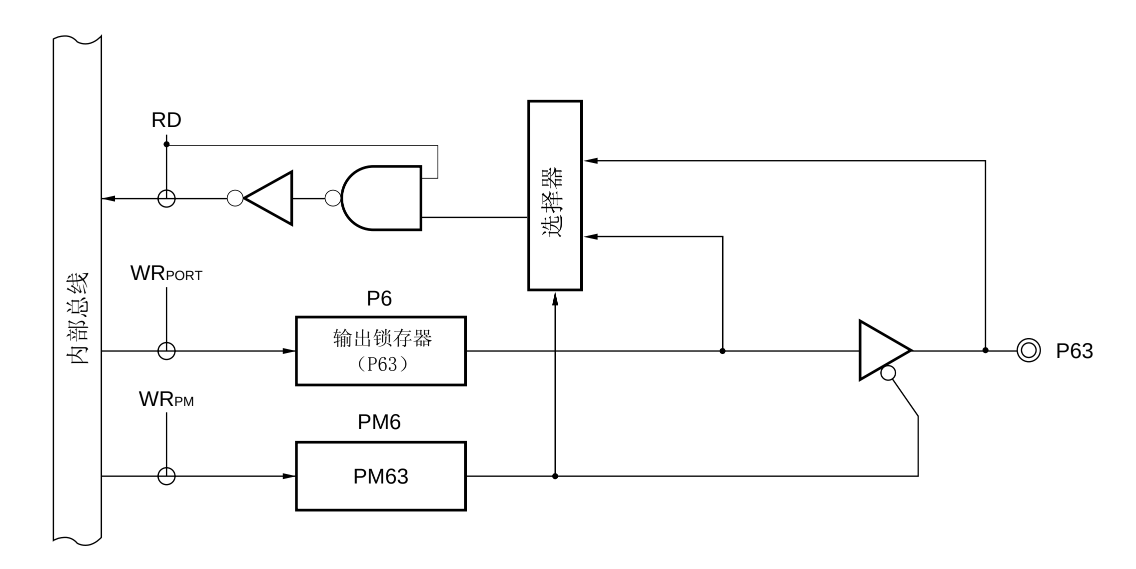
WR××:　写信号

图 **4-14. P62** 的框图




P6: 端口寄存器 6
PM6: 端口模式寄存器 6
RD: 读信号
WR××: 写信号

图 **4-15. P63** 的框图




P6: 端口寄存器 6
PM6: 端口模式寄存器 6
RD: 读信号
WR××: 写信号

**4.2.7 端口 7**

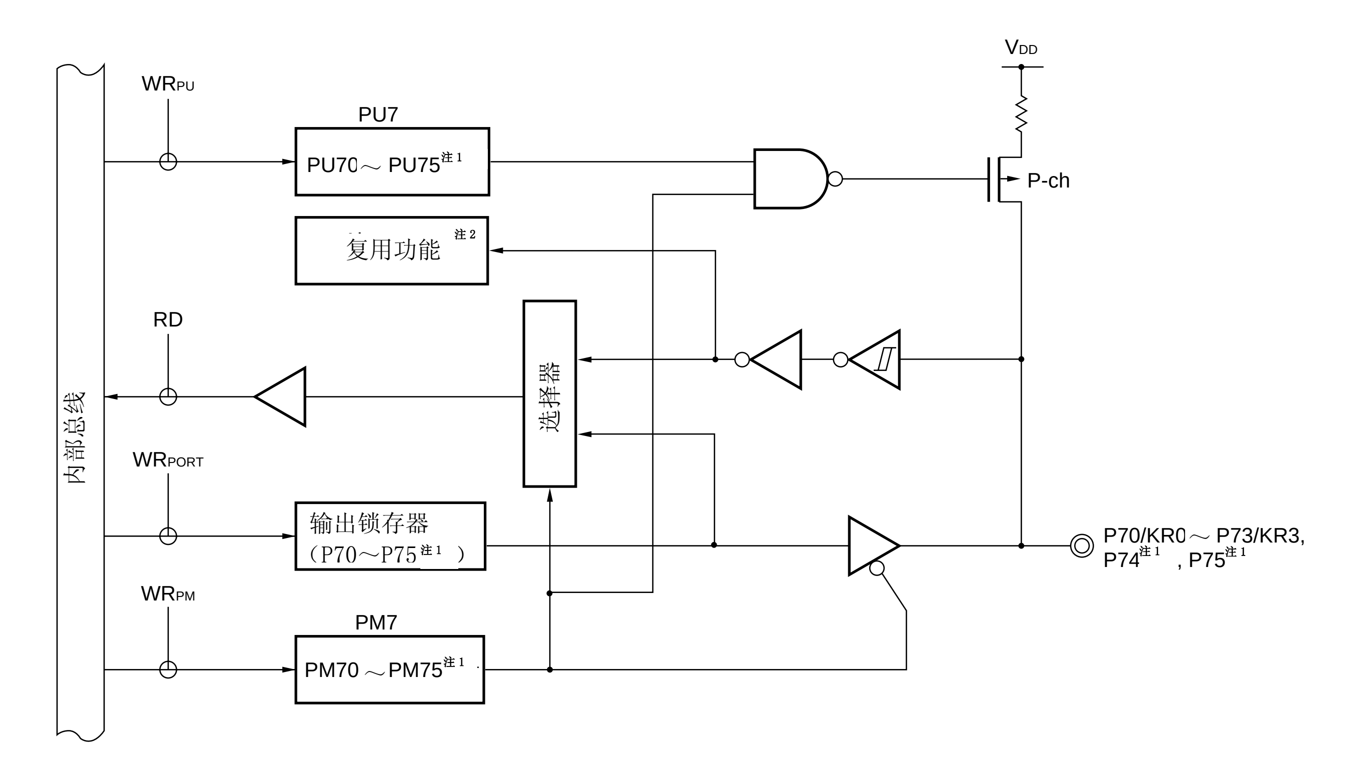
端口 7 是具有输出锁存功能的 6 位 I/O 端口。通过使用端口模式寄存器 7(PM7)，可以位选端口 7 为输入或输出模式。当 P70 ～ P75 作为输入端口时，内部上拉电阻的使用可以通过上拉电阻选择寄存器 7(PU7)以 1 位单元的方式指定。

P70 ～ P73 还可以用作按键返回输入。

复位信号的产生可将端口 7 设置为输入模式。

图 4-16 显示端口 7 的框图。

**图 4-16. P70 ～ P75 的框图**




注　**1.** P74, P75, PM74, PM75, PU74, 和 PU75 仅在 48 引脚产品中可用。

**2.** 只有在 P70 ～ P73 引脚可用复用功能。

P7:　端口寄存器 7

PU7:　上拉电阻选择寄存器 7

PM7:　端口模式寄存器 7

RD:　读信号

WR××:　写信号

### 4.2.8 端口 12

端口 12 是具有输出锁存功能的 5 位 I/O 端口。通过使用端口模式寄存器 12(PM12)，可以位选端口 12 为输入或输出模式。当 P120 只用作输入端口时，内部上拉电阻的使用可以通过上拉电阻选择寄存器 12(PU12)指定。

这个端口还可以用于外部中断请求输入、外部低电压检测的电压输入、主系统时钟振荡器的连接、副时钟振荡器的连接、主系统时钟的外部时钟输入和副时钟的外部时钟输入。

复位信号的产生可将端口 12 设置为输入模式。

图 4-17 和 4-18 显示端口 12 的框图。

**注意事项 1. 当使用 P121 ～ P124 引脚连接主系统时钟（X1,X2）或副时钟（XT1,XT2）的振荡器，或者输入该主系统时钟的外部时钟（EXCLK）或副时钟的外部时钟（EXCLKS）时，必须通过使用时钟操作模式选择寄存器（OSCCTL）对 X1 振荡模式、XT1 振荡模式或者外部时钟输入模式进行设置（如需了解详细信息，可参见 5.3 (1) 时钟操作模式选择寄存器（OSCCTL）和（3）副时钟引脚操作模式设置）。OSCCTL 的复位值为 00H（P121 ～ P124 都用作 I/O 端口引脚）。在这种情况下，不必对 PM121 ～ PM124 和 P121 ～ P124 引脚进行设置。**

<R> **2. 对于具有 48KB或更大Flash存储器而没有片上调试功能的产品 (μPD78F0514 和 78F0515)，有一个 “I”, “K”, 或 “E” 的产品等级，对于具有片上调试功能的产品 (μPD78F0513D 和 78F0515D) 在使用Flash编程器写Flash存储器时，应按以下连接P121/X1/OCD0A[注]。**

**• P121/X1/OCD0A[注]： 通过电阻连接到 $V_{SS}$ (10 kΩ: 推荐)(在输入模式)或悬空(在输出模式)。**

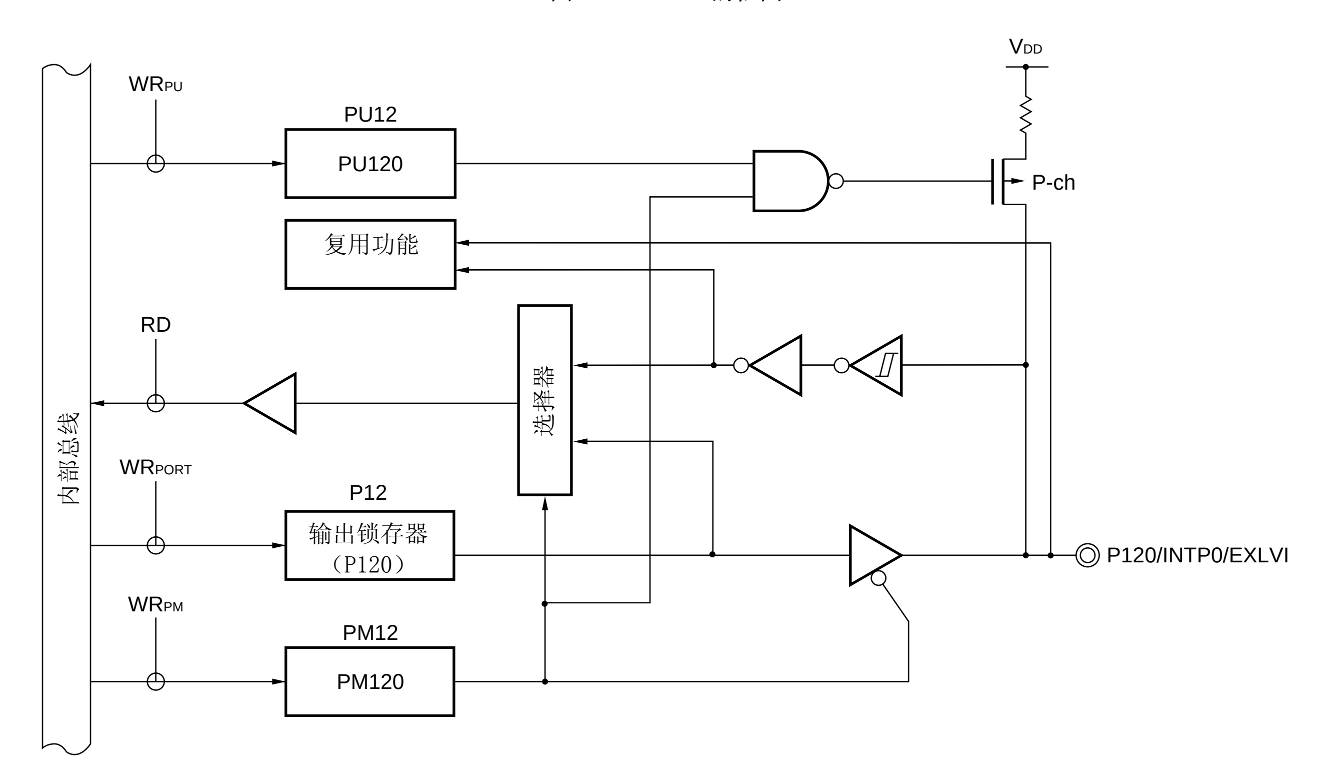
**当通过自编程方式写入 Flash 存储器时，以上连接并不需要。**

<R> 注　只有μPD78F0513D 和 78F0515D 提供 OCD0A 。

<R> 备注 1. 对于产品级别，请咨询 NEC 电子的销售代表。

2. μPD78F0513D 和 78F0515D 在使用片上调试功能时，X1 和 X2 能够用于片上调试模式设定引脚 (OCD0A, OCD0B) 。详情请参看 **第 26 章 片上调试功能 (仅μPD78F0513D 和 78F0515D)** 。

图 **4-17. P120** 的框图




P12:　端口寄存器 12
PU12:　上拉电阻选择寄存器 12
PM12:　端口模式寄存器 12
RD:　读信号
WR××:　写信号

图 4-18. P121 ～ P124 的框图

P12:　　端口寄存器 12

PU12:　　上拉电阻选择寄存器 12

PM12:　　端口模式寄存器 12

OSCCTL:时钟操作模式选择寄存器

RD:　　读信号

WR××:　　写信号

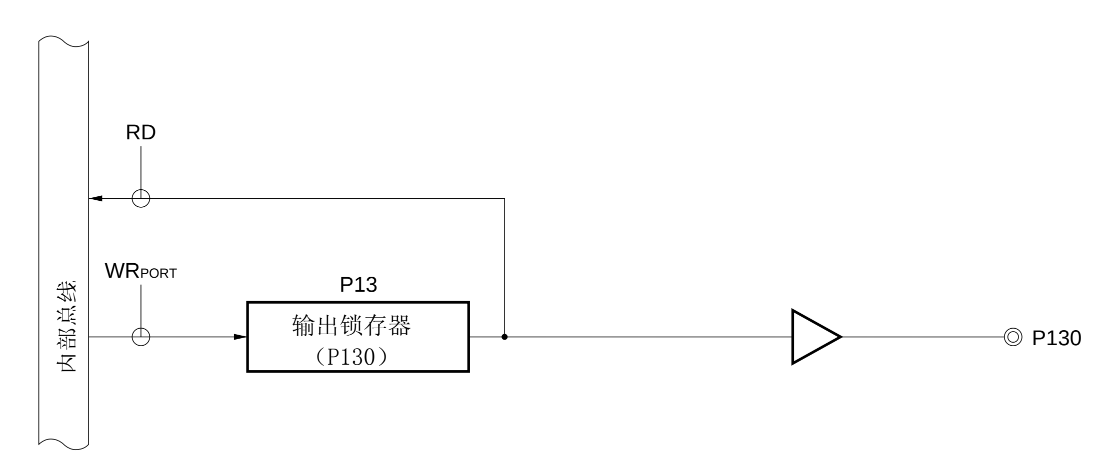
注　　仅用于 μPD78F0513D 和 78F0515D 。

**4.2.9 端口 13 (仅 48 引脚产品)**

端口 13 是 1 位仅输出端口。

图 4-19 显示了端口 13 的框图。

**图 4-19. P130 的框图**

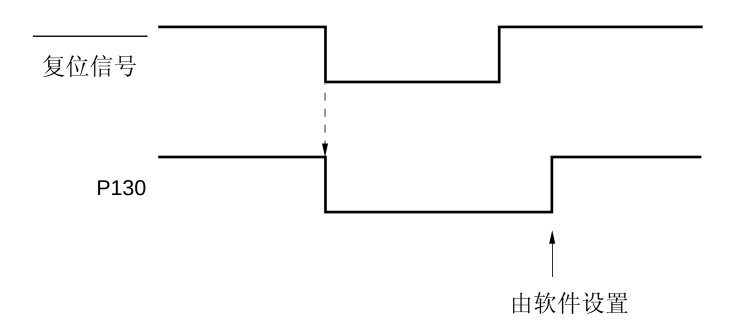



P13: 端口寄存器 13

RD: 读信号

WR××: 写信号

**备注** 当复位信号有效时，P130 输出低电平。如果在复位信号有效前将 P130 设置为高电平输出，则 P130 的输出信号可以作为虚拟的 CPU 复位信号。



### 4.2.10 端口 14 (仅 48 引脚产品)

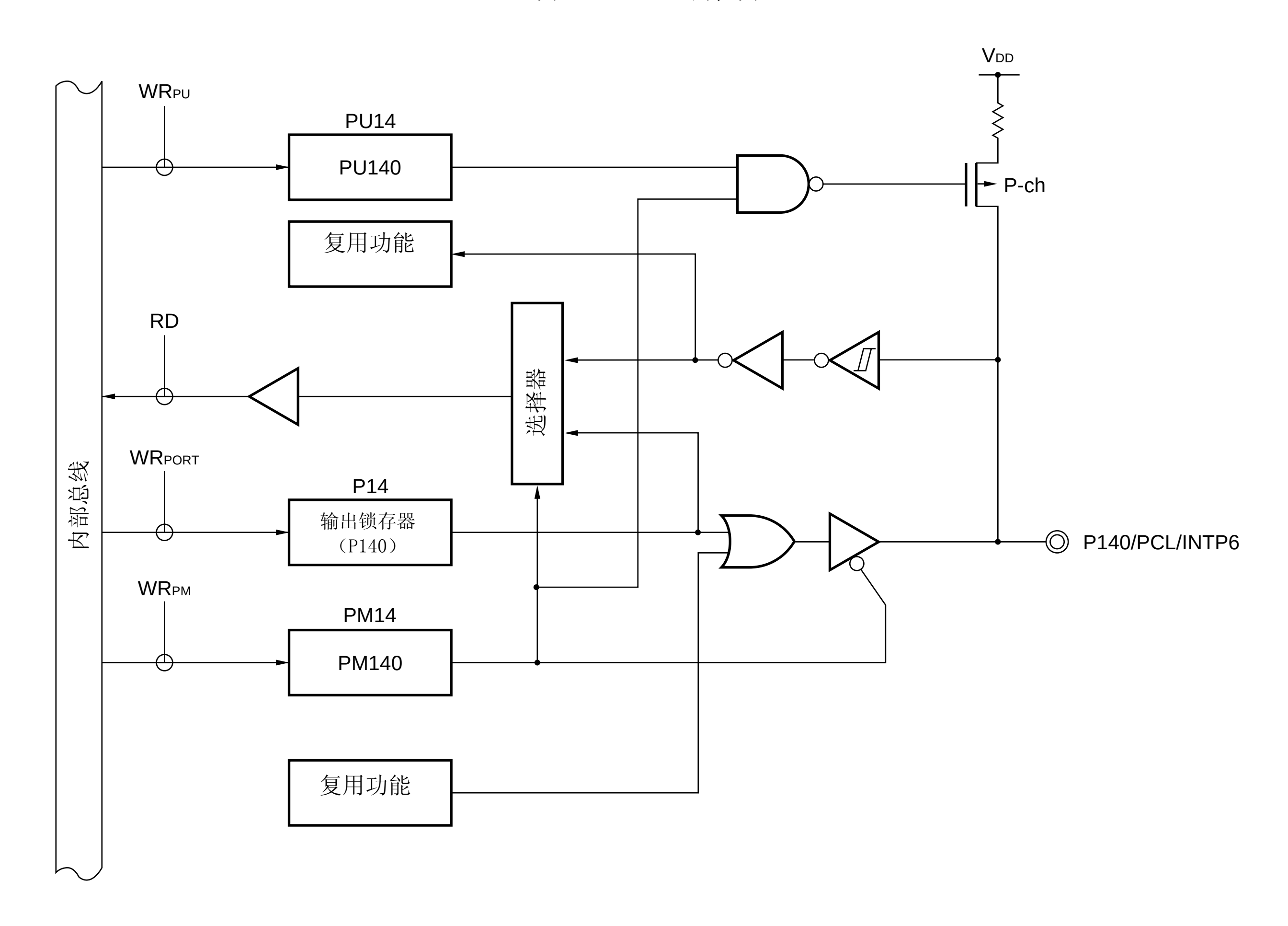
端口 14 是具有输出锁存功能的 1 位 I/O 端口。通过使用端口模式寄存器 14(PM14)，可以位选端口 14 为输入或输出模式。当 P140 作为输入端口时，内部上拉电阻的使用可以通过上拉电阻选择寄存器 14(PU14)以 1 位单元的方式指定。

这个端口还可以用于外部中断请求输入和时钟输出。

复位信号的产生可将端口 14 设置为输入模式。

图 4-20 显示端口 14 的框图。

**图 4-20. P140 的框图**




P14:　端口寄存器 14

PU14:　上拉电阻选择寄存器 14

PM14:　端口模式寄存器 14

RD:　读信号

WR××:　写信号

## 4.3 控制端口功能的寄存器

端口功能由如下四种类型寄存器控制：

- 端口模式寄存器 (PM0 ～ PM4, PM6, PM7, PM12, PM14[注])
- 端口寄存器 (P0 ～ P4, P6, P7, P12, P13[注], P14[注])
- 上拉电阻选择寄存器 (PU0, PU1, PU3, PU4, PU7, PU12, PU14[注])
- A/D 端口配置寄存器 (ADPC)

注　仅限 48 引脚产品。

### (1) 端口模式寄存器 (PM0 ～ PM4, PM6, PM7, PM12, 和 PM14[注])

这类寄存器以 1 位单元的方式定义端口的输入或输出模式。
可以由 1 位或 8 位存储器操作指令设置这些寄存器。
复位信号的产生可将这些寄存器的内容设置为 FFH。
当端口引脚使用复用功能时，需要参考 **4.5 使用复用功能时端口模式寄存器和输出锁存器的设置**，对端口模式寄存器进行设置。

注　仅限 48 引脚产品。

图 **4-21.** 端口模式寄存器的格式

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 地址 | 复位值 | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PM0 | 1 | 1 | 1 | 1 | 1 | 1 | PM01 | PM00 | FF20H | FFH | R/W |
| PM1 | PM17 | PM16 | PM15 | PM14 | PM13 | PM12 | PM11 | PM10 | FF21H | FFH | R/W |
| PM2 | PM27 | PM26 | PM25 | PM24 | PM23 | PM22 | PM21 | PM20 | FF22H | FFH | R/W |
| PM3 | 1 | 1 | 1 | 1 | PM33 | PM32 | PM31 | PM30 | FF23H | FFH | R/W |
| PM4 | 1 | 1 | 1 | 1 | 1 | 1 | PM41 | PM40 | FF24H | FFH | R/W |
| PM6 | 1 | 1 | 1 | 1 | PM63 | PM62 | PM61 | PM60 | FF26H | FFH | R/W |
| PM7 | 1 | 1 | PM75[注] | PM74[注] | PM73 | PM72 | PM71 | PM70 | FF27H | FFH | R/W |
| PM12 | 1 | 1 | 1 | PM124 | PM123 | PM122 | PM121 | PM120 | FF2CH | FFH | R/W |
| PM14[注] | 1 | 1 | 1 | 1 | 1 | 1 | 1 | PM140 | FF2EH | FFH | R/W |

| PMmn | Pmn 引脚 I/O 模式选择<br>(m = 0 ～ 4, 6, 7, 12, 14; n = 0 ～ 7) |
|---|---|
| 0 | 输出模式 (输出缓冲器打开) |
| 1 | 输入模式 (输出缓冲器关闭) |

注　仅限 48 引脚产品。

<R>　**注意事项**　对于 **44** 引脚产品，请设定 **PM0** 的 **2** 到 **7** 位、**PM3** 的 **4** 到 **7** 位、**PM4** 的 **2** 到 **7** 位、**PM6** 的 **4** 到 **7** 位、**PM7** 的 **4** 到 **7** 位和 **PM12** 的 **5** 到 **7** 位为 **1**。
对于 **48** 引脚产品，请设定 **PM0** 的 **2** 到 **7** 位、**PM3** 的 **4** 到 **7** 位、**PM4** 的 **2** 到 **7** 位、**PM6** 的 **4** 到 **7** 位、**PM7** 的 **6** 到 **7** 位和 **PM12** 的 **5** 到 **7** 位和 **PM14** 的 **1** 到 **7** 位为 **1**。

**(2) 端口寄存器 (P0 ～ P4, P6, P7, P12, P13注, 和 P14注)**

这类寄存器用于设置芯片端口要输出的数据。

如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值。

可以由 1 位或 8 位存储器操作指令设置这些寄存器。

复位信号的产生可将寄存器清零(00H)。

注　　仅限 48 引脚产品。

**图 4-22. 端口寄存器的格式**

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 地址 | 复位值 | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| P0 | 0 | 0 | 0 | 0 | 0 | 0 | P01 | P00 | FF00H | 00H ( 输出锁存 ) | R/W |
| P1 | P17 | P16 | P15 | P14 | P13 | P12 | P11 | P10 | FF01H | 00H ( 输出锁存 ) | R/W |
| P2 | P27 | P26 | P25 | P24 | P23 | P22 | P21 | P20 | FF02H | 00H ( 输出锁存 ) | R/W |
| P3 | 0 | 0 | 0 | 0 | P33 | P32 | P31 | P30 | FF03H | 00H ( 输出锁存 ) | R/W |
| P4 | 0 | 0 | 0 | 0 | 0 | 0 | P41 | P40 | FF04H | 00H ( 输出锁存 ) | R/W |
| P6 | 0 | 0 | 0 | 0 | P63 | P62 | P61 | P60 | FF06H | 00H ( 输出锁存 ) | R/W |
| P7 | 0 | 0 | P75注 1 | P74注 1 | P73 | P72 | P71 | P70 | FF07H | 00H ( 输出锁存 ) | R/W |
| P12 | 0 | 0 | 0 | P124注 2 | P123注 2 | P122注 2 | P121注 2 | P120 | FF0CH | 00H ( 输出锁存 ) | R/W |
| P13注 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | P130 | FF0DH | 00H ( 输出锁存 ) | R/W |
| P14注 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | P140 | FF0EH | 00H ( 输出锁存 ) | R/W |

| Pmn | m = 0 ～ 4, 6, 7, 12 ～ 14; n = 0 ～ 7 | |
|---|---|---|
| | 输出数据控制(输出模式下) | 输入数据读取 (输入模式下) |
| 0 | 输出 0 | 输入低电平 |
| 1 | 输出 1 | 输入高电平 |

注　**1.** 仅限 48 引脚产品。

<R>　**2.** 当 P121 ～ P124 在外部时钟输入模式下，读取它们的输出锁存器总是 “0” 。

**(3) 上拉电阻选择寄存器 (PU0, PU1, PU3, PU4, PU7, PU12, 和 PU14[注])**

这类寄存器指定是否使用P00, P01, P10 ～ P17, P30 ～ P33, P40, P41, P70 ～ P73, P74[注], P75[注], P120, 和 P140[注]的内部上拉电阻。内部上拉电阻可以以 1 位单元的方式用于那些已设置为输入模式的引脚，并且这些引脚已通过PU0, PU1, PU3, PU4, PU7, PU12, 和 PU14[注]指定为使用内部上拉电阻。不论是否设置了PU0, PU1, PU3, PU4, PU7, PU12, 和 PU14[注]，设置为输出模式的引脚和用作复用功能的输出引脚都不连接内部上拉电阻。

可以由 1 位或 8 位存储器操作指令设置这些寄存器。

复位信号的产生可将这些寄存器清零(00H)。

注　　仅限 48 引脚产品。

**图 4-23. 上拉电阻选择寄存器的格式**

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 地址 | 复位值 | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PU0 | 0 | 0 | 0 | 0 | 0 | 0 | PU01 | PU00 | FF30H | 00H | R/W |
| PU1 | PU17 | PU16 | PU15 | PU14 | PU13 | PU12 | PU11 | PU10 | FF31H | 00H | R/W |
| PU3 | 0 | 0 | 0 | 0 | PU33 | PU32 | PU31 | PU30 | FF33H | 00H | R/W |
| PU4 | 0 | 0 | 0 | 0 | 0 | 0 | PU41 | PU40 | FF34H | 00H | R/W |
| PU7 | 0 | 0 | PU75[注] | PU74[注] | PU73 | PU72 | PU71 | PU70 | FF37H | 00H | R/W |
| PU12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PU120 | FF3CH | 00H | R/W |
| PU14[注] | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PU140 | FF3EH | 00H | R/W |

| PUmn | Pmn 引脚内部上拉电阻选择<br>(m = 0, 1, 3, 4, 7, 12, 14; n = 0 ～ 7) |
|---|---|
| 0 | 不连接内部上拉电阻 |
| 1 | 连接内部上拉电阻 |

注　　仅限 48 引脚产品。

**(4) A/D 端口配置寄存器 (ADPC)**

这个寄存器将 P20/ANI0 ～ P27/ANI7 引脚切换为数字 I/O 端口 或 A/D 转换器的模拟输入。

可由 1 位或者 8 位存储器操作指令对 ADPC 进行设置。

复位信号的产生可将寄存器清零(00H)。

**图 4-24. A/D 端口配置寄存器 (ADPC) 的格式**

地址 ：FF2FH　复位后 ：00H　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADPC | 0 | 0 | 0 | 0 | ADPC3 | ADPC2 | ADPC1 | ADPC0 |

| ADPC3 | ADPC2 | ADPC1 | ADPC0 | 数字 I/O (D)/ 模拟输入 (A) 转换 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | P27/ANI7 | P26/ANI6 | P25/ANI5 | P24/ANI4 | P23/ANI3 | P22/ANI2 | P21/ANI1 | P20/ANI0 |
| 0 | 0 | 0 | 0 | A | A | A | A | A | A | A | A |
| 0 | 0 | 0 | 1 | A | A | A | A | A | A | A | D |
| 0 | 0 | 1 | 0 | A | A | A | A | A | A | D | D |
| 0 | 0 | 1 | 1 | A | A | A | A | A | D | D | D |
| 0 | 1 | 0 | 0 | A | A | A | A | D | D | D | D |
| 0 | 1 | 0 | 1 | A | A | A | D | D | D | D | D |
| 0 | 1 | 1 | 0 | A | A | D | D | D | D | D | D |
| 0 | 1 | 1 | 1 | A | D | D | D | D | D | D | D |
| 1 | 0 | 0 | 0 | D | D | D | D | D | D | D | D |
| 除此之外 | | | | 禁止设置 | | | | | | | |

注意事项　**1.** 通过使用端口模式寄存器 **2(PM2)**将 **A/D** 转换通道设置为输入模式。

**2.** 如果将数据写入 **ADPC**，则产生一个等待周期。当 **CPU** 使用子系统时钟并且外部硬件时钟停止时，不要将数据写入 **ADPC**。详细情况可参见 第三十二章 等待注意事项。

## 4.4 端口功能操作

对输入模式和输出模式的端口操作是不同的，具体如下所示。

### 4.4.1 写入 I/O 端口

**(1) 输出模式**

使用传送指令对输出锁存器进行写操作，输出锁存器的内容从引脚输出。
一旦数据写入输出锁存器，它将一直保存到新数据被写入。
当复位信号产生时输出锁存器的内容被清零。

**(2) 输入模式**

使用传送指令对输出锁存器进行写操作，因为输出缓冲器处于关闭状态，所以引脚状态不会改变。
一旦数据写入输出锁存器，它将一直保存到新数据被写入。
当复位信号产生时输出锁存器的内容被清零。

### 4.4.2 读取 I/O 端口

**(1) 输出模式**

使用传送指令读取的是输出锁存器的内容。输出锁存器的内容不会改变。

**(2) 输入模式**

使用传送指令读取的是引脚状态。输出锁存器的内容不会改变。

### 4.4.3 I/O 端口的操作

**(1) 输出模式**

对输出锁存器执行一个操作时，操作结果写入输出锁存器。而输出锁存器的内容则从引脚输出。
一旦数据写入输出锁存器，它将一直保存到新数据被写入。
当复位信号的产生时输出锁存器的内容被清零。

**(2) 输入模式**

读取引脚电平，对它的内容执行操作，操作的结果写入输出锁存器，因为输出缓冲器处于关闭状态，所以引脚状态不会改变。
当复位信号的产生时输出锁存器的内容被清零。

## 4.5 使用复用功能时端口模式寄存器和输出锁存器的设置

为使用端口引脚的复用功能，应按 表 4-5 所示对端口模式寄存器和输出锁存器进行设置。

**表 4-5. 使用复用功能时，端口模式寄存器和输出锁存器的设置 (1/2)**

| 引脚名称 | 复用功能 | | PM×× | P×× |
|---|---|---|---|---|
| | 功能名称 | I/O | | |
| P00 | TI000 | 输入 | 1 | × |
| P01 | TI010 | 输入 | 1 | × |
| | TO00 | 输出 | 0 | 0 |
| P10 | $\overline{SCK10}$ | 输入 | 1 | × |
| | | 输出 | 0 | 1 |
| | TxD0 | 输出 | 0 | 1 |
| P11 | SI10 | 输入 | 1 | × |
| | RxD0 | 输入 | 1 | × |
| P12 | SO10 | 输出 | 0 | 0 |
| P13 | TxD6 | 输出 | 0 | 1 |
| P14 | RxD6 | 输入 | 1 | × |
| P15 | TOH0 | 输出 | 0 | 0 |
| P16 | TOH1 | 输出 | 0 | 0 |
| | INTP5 | 输入 | 1 | × |
| P17 | TI50 | 输入 | 1 | × |
| | TO50 | 输出 | 0 | 0 |
| P20 ～ P27[注 1] | ANI0 ～ ANI7[注 1] | 输入 | 1 | × |
| P30 ～ P32 | INTP1 ～ INTP3 | 输入 | 1 | × |
| P33 | INTP4 | 输入 | 1 | × |
| | TI51 | 输入 | 1 | × |
| | TO51 | 输出 | 0 | 0 |
| P60 | SCL0 | I/O | 0 | 0 |
| P61 | SDA0 | I/O | 0 | 0 |
| P62 | EXSCL0 | 输入 | 1 | × |
| P70 ～ P73 | KR0 ～ KR3 | 输入 | 1 | × |
| P120 | INTP0 | 输入 | 1 | × |
| | EXLVI | 输入 | 1 | × |
| P121 | X1[注 2] | – | × | × |
| P122 | X2[注 2] | – | × | × |
| | EXCLK[注 2] | 输入 | × | × |
| P123 | XT1[注 2] | – | × | × |
| P124 | XT2[注 2] | – | × | × |
| | EXCLKS[注 2] | 输入 | × | × |
| P140[注 3] | PCL | 输出 | 0 | 0 |
| | INTP6 | 输入 | 1 | × |

(注 1 ～ 3, 备注 1 和 2 下页列出)

注　**1.**　通过使用 A/D 端口配置寄存器 (ADPC) 、模拟输入通道规格寄存器(ADS)和 PM2, 可以选择 ANI0/P20 ～ ANI7/P27 引脚的功能。

<R>

**表 4-6. ANI0/P20 ～ ANI7/P27** 引脚的设定功能

| ADPC | PM2 | ADS | ANI0/P20 ～ ANI7/P27 引脚 |
|---|---|---|---|
| 模拟输入选择 | 输入模式 | 选择 ANI. | 模拟输入 (转换) |
| | | 不选择 ANI. | 模拟输入 (不转换) |
| | 输出模式 | 选择 ANI. | 禁止设置 |
| | | 不选择 ANI. | |
| 数字 I/O 选择 | 输入模式 | – | 数字输入 |
| | 输出模式 | – | 数字输出 |

**2.**　当使用 P121 ～ P124 引脚连接主系统时钟（X1,X2）或副时钟（XT1,XT2）的振荡器，或者输入该主系统时钟的外部时钟（EXCLK）或副时钟的外部时钟（EXCLKS）时，必须通过使用时钟操作模式选择寄存器（OSCCTL）对 X1 振荡模式、XT1 振荡模式或者外部时钟输入模式进行设置（如需了解详细信息，可参见 **5.3 (1) 时钟操作模式选择寄存器（OSCCTL）**和（**3**）**副时钟引脚操作模式设置**）。OSCCTL 的复位值为 00H（P121 ～ P124 都用作 I/O 端口引脚）。在这种情况下，不必对 PM121 ～ PM124 和 P121 ～ P124 引脚进行设置。

**3.**　仅限 48 引脚产品。

备注　**1.** ×:　不必考虑

PM××:　端口模式寄存器

P××:　端口输出锁存器

**2.**　μPD78F0513D 和 78F0515D 在使用片上调试功能时，X1、 X2、 P31 和 P32 能够用于片上调试模式设定引脚 (OCD0A, OCD0B, OCD1A, OCD1B) 。详情请参看 **第二十六章　片上调试功能　(仅 μPD78F0513D 和 78F0515D)** 。

<R>

## 4.6 对端口寄存器 n (Pn) 的 1 位处理指令的注意事项

当对一个支持输入输出功能的端口执行一个 1 位处理指令时，一个输入端口的输出锁存器的值（与目标位不同）可能被写入。

因此，建议一个端口从输入模式切换为输出模式时重写输出锁存器。

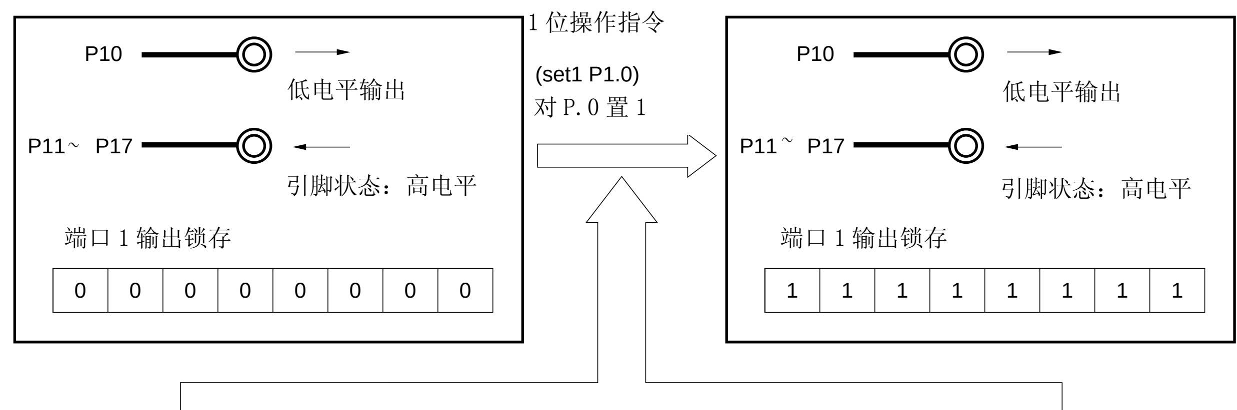
<例子>　当 P10 是输出端口， P11 ～ P17 是输入端口 (所有引脚的状态都是高电平)，端口 1 的锁存器值是 00H，如果通过 1 位处理指令，端口 10 输出从低电平变为高电平， 则端口 1 的输出锁存器值为 FFH。

解释:　有时，PMnm 位为 1 的 Pn 寄存器读写的目标是输出锁存器和引脚状态。
78K0/KC2 产品按照以下步骤执行 1 位操作指令。

<1> Pn 寄存器按 8 位读取。
<2> 只处理目标位。
<3> 以 8 位写入 Pn 寄存器。

在步骤 <1>，当作为输入端口的 P11 ～ P17 的引脚状态被读取时，读取作为输出端口的 P10 输出锁存器(0)的值。如果 P11 ～ P17 引脚状态为高电平时，读取值为 FEH。
通过 <2> 中将值变为 FFH 。
通过 <3> 将 FFH 写入输出锁存器中。

**图 4-25. 位处理指令 (P10)**




对 P10 位的 1 位操作指令

<1> 端口寄存器 P1 以 8 位为单位进行读操作。
- 端口为 P10 的情况下，输出端口，读端口输出锁存器（0）的数值。
- 端口为 P11 到 P17 的情况下，输入端口，读引脚状态值（1）。

<2> 设置 P10 位为 1。
<3> 将<2>的结果以 8 位为单位写入端口寄存器 1（P1）中。

# 第五章　时钟发生器

## 5.1 时钟发生器的功能

时钟发生器用于产生时钟提供给 CPU 和外部硬件设备。
可以使用以下三种系统时钟和时钟振荡器。

**(1)** 主系统时钟

**<1> X1 振荡器**

通过连接一个振荡器到 X1 和 X2，该振荡电路产生 $f_X$ =1 到 20MHz 的时钟。
通过执行 STOP 指令或设置主 OSC 控制寄存器(MOC)，可以停止振荡。

**<2> 内部高速振荡器**

这个振荡电路产生一个 $f_{RH}$ = 8 MHz (TYP.)的时钟。复位释放后，CPU 总是使用这个内部高速振荡时钟进行操作。可以通过执行 STOP 指令或者使用内部振荡模式寄存器(RCM)停止其振荡。

外部主系统时钟($f_{EXCLK}$ = 1 到 20 MHz)也可以通过 EXCLK/X2/P122 引脚提供。可以通过执行 STOP 指令或者使用 RCM 禁止外部主系统时钟输入。
可以通过使用主时钟模式寄存器(MCM)选择内部高速系统时钟(X1 时钟或者外部主系统时钟)或内部高速振荡时钟，作为主系统时钟。

**(2)** 副系统时钟

- **副系统时钟振荡器**

  通过在 XT1 和 XT2 之间连接一个 32.768kHz 的振荡器，该电路以 $f_{XT}$ = 32.768 kHz 的频率进行振荡。通过使用处理器时钟控制寄存器(PCC)和时钟操作模式选择寄存器(OSCCTL)，可以停止振荡。

还可以通过 EXCLKS/XT2/P124 引脚提供一个外部副系统时钟 ($f_{EXCLKS}$ = 32.768 kHz)。可以通过设置 PCC 和 OSCCTL 禁止外部副系统时钟输入。

备注　**1.** $f_X$:　X1 时钟振荡频率
　　　**2.** $f_{RH}$:　内部高速振荡时钟频率
　　　**3.** $f_{EXCLK}$:　外部主系统时钟频率
　　　**4.** $f_{XT}$:　XT1 时钟振荡频率
　　　**5.** $f_{EXCLKS}$:　外部副系统时钟频率

**(3) 内部低速振荡时钟(看门狗定时器时钟)**

- **内部低速振荡器**

  该电路以 $f_{RL}$ = 240 kHz (TYP.)的时钟振荡。复位释放后，内部低速振荡时钟总是启动操作。

  当通过选项字节设置“内部低速振荡器可由软件停止”时，可以通过使用内部振荡模式寄存器(RCM)停止其振荡。

  内部低速振荡时钟不能作为 CPU 时钟。如下硬件使用内部低速振荡时钟。

  - 看门狗定时器
  - TMH1 (选择 $f_{RL}$，$f_{RL}/2^7$或 $f_{RL}/2^9$时)

备注　$f_{RL}$:　内部低速振荡时钟频率

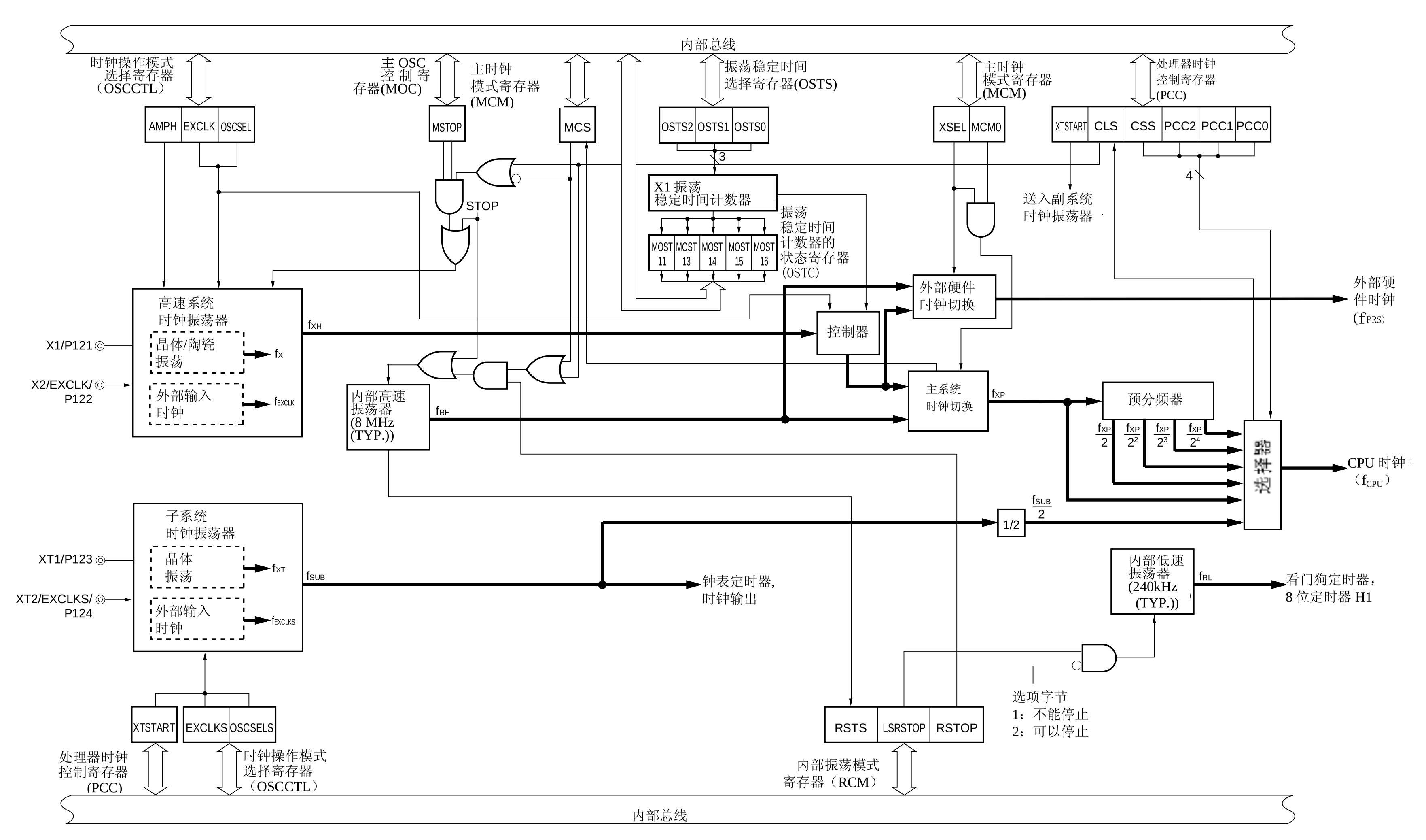
## 5.2 时钟发生器的构成

时钟发生器包括以下硬件。

**表 5-1. 时钟发生器的构成**

| 项目 | 构成 |
| --- | --- |
| 控制寄存器 | 时钟操作模式选择寄存器(OSCCTL)<br>处理器时钟控制寄存器 (PCC)<br>内部振荡模式寄存器 (RCM)<br>主 OSC 控制寄存器 (MOC)<br>主时钟模式寄存器 (MCM)<br>振荡稳定时间计数器的状态寄存器(OSTC)<br>振荡稳定时间选择寄存器(OSTS) |
| 振荡器 | X1 振荡器<br>XT1 振荡器<br>内部高速振荡器<br>内部低速振荡器 |

<R>图 5-1 时钟发生器的框图



备注　**1.** $f_X$:　X1 时钟振荡频率
**2.** $f_{RH}$:　内部高速振荡时钟频率
**3.** $f_{EXCLK}$:　外部主系统时钟频率
**4.** $f_{XH}$:　高速系统时钟频率
**5.** $f_{XP}$:　主系统统时钟频率
**6.** $f_{PRS}$:　外部硬件时钟频率
**7.** $f_{CPU}$:　CPU 时钟频率
**8.** $f_{XT}$:　XT1 时钟振荡频率
**9.** $f_{EXCLKS}$:　外部副系统时钟频率
**10.** $f_{SUB}$:　副系统时钟频率
**11.** $f_{RL}$:　内部低速振荡时钟频率

## 5.3 控制时钟发生器的寄存器

以下七种寄存器用于控制时钟发生器。

- 时钟操作模式选择寄存器 (OSCCTL)
- 处理器时钟控制寄存器 (PCC)
- 内部振荡模式寄存器 (RCM)
- 主 OSC 控制寄存器(MOC)
- 主时钟模式寄存器 (MCM)
- 振荡稳定时间计数器的状态寄存器 (OSTC)
- 振荡稳定时间选择寄存器 (OSTS)

**(1) 时钟操作模式选择寄存器 (OSCCTL)**

该寄存器用于选择高速系统和副系统时钟的操作模式，以及内置振荡器的获取方式。
可由一个 1 位或 8 位存储器操作指令设置 OSCCTL。
复位信号的产生可将这个寄存器设置为 00H。

图 **5-2.** 时钟操作模式选择寄存器的格式 **(OSCCTL)**

地址: FF9FH　复位后: 00H　R/W

| 符号 | <7> | <6> | <5> | <4> | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| OSCCTL | EXCLK | OSCSEL | EXCLKS[注] | OSCSELS[注] | 0 | 0 | 0 | AMPH |

| EXCLK | OSCSEL | 高速系统时钟引脚操作模式 | P121/X1 引脚 | P122/X2/EXCLK 引脚 |
|---|---|---|---|---|
| 0 | 0 | I/O 端口模式 | I/O 端口 | |
| 0 | 1 | X1 振荡模式 | 晶体/陶瓷振荡器连接 | |
| 1 | 0 | I/O 端口模式 | I/O 端口 | |
| 1 | 1 | 外部时钟输入模式 | I/O 端口 | 外部时钟输入 |

| AMPH | 操作频率控制 |
|---|---|
| 0 | 1 MHz ≤ $f_{XH}$ ≤ 10 MHz |
| 1 | 10 MHz < $f_{XH}$ ≤ 20 MHz |

注　EXCLKS 和 OSCSELS 与 XTSTART(处理器时钟控制寄存器 (PCC)的第 6 位)结合使用。参见 **(3)** 副系统时钟引脚操作模式的设置。

注意事项
**1.** 如果高速系统时钟振荡频率超过 **10MHz**，则必须将 **AMPH** 设置为 **1**。

<R> **2.** 复位释放后外部功能设置之前应设置 **AMPH**。复位释放后 **AMPH** 的值只能被修改一次。当高速系统时钟（**X1** 振荡）被选为 **CPU** 时钟，在 **AMPH** 被设为 **1** 之后要提供给 **CPU** 一个 **4.06** 到 **16.12us** 的时钟停止期。当高速系统时钟（外部时钟输入）被选为 **CPU** 时钟，在 **AMPH** 被设为 **1** 之后要提供给 **CPU160** 个外部时钟信号周期的始终停止期。

<R> **3.** 当内部高速振荡时钟作为 **CPU** 时钟使用且 **AMPH=1** 时，执行 **STOP** 指令，则在 **STOP** 模式释放后，要提供给 **CPU** 一个 **4.06** 到 **16.12us** 的时钟停止期。如果或者外部主系统时钟作为 **CPU** 时钟（外部时钟输入），则要提供给 **CPU** 一个为期 **160** 个外部时钟周期的停止期。如果 **X1** 时钟用作 **CPU** 时钟，振荡稳定时间从 **STOP** 模式释放后开始计算。

**4.** 若要修改 **EXCLK** 和 **OSCSEL**，必须确保主 **OSC** 控制寄存器**(MOC)**的第 **7** 位**(MSTOP) =1(X1** 振荡器停止或禁止使用来自 **EXCLK** 引脚的外部时钟**)**。

备注　$f_{XH}$: 高速系统时钟振荡频率

**(2)　处理器时钟控制寄存器(PCC)**

PCC 用于选择 CPU 时钟、分频比和副系统时钟的操作模式。

由 1 位或 8 位存储器操作指令设置 PCC。

复位信号的产生可将 PCC 设置为 01H。

**图 5-3.　处理器时钟控制寄存器(PCC)的格式**

地址: FFFBH　复位后: 01H　R/W[注 1]

| 符号 | 7 | 6 | <5> | <4> | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PCC | 0 | XTSTART[注 2] | CLS | CSS | 0 | PCC2 | PCC1 | PCC0 |

| CLS | CPU 时钟状态 |
|---|---|
| 0 | 主系统时钟 |
| 1 | 副系统时钟 |

| CSS | PCC2 | PCC1 | PCC0 | CPU 时钟 ($f_{CPU}$) 选择 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $f_{XP}$ |
| | 0 | 0 | 1 | $f_{XP}/2$ (默认) |
| | 0 | 1 | 0 | $f_{XP}/2^2$ |
| | 0 | 1 | 1 | $f_{XP}/2^3$ |
| | 1 | 0 | 0 | $f_{XP}/2^4$ |
| 1 | 0 | 0 | 0 | $f_{SUB}/2$ |
| | 0 | 0 | 1 | |
| | 0 | 1 | 0 | |
| | 0 | 1 | 1 | |
| | 1 | 0 | 0 | |
| 其他情况 | | | | 禁止设置 |

注　**1.** 第 5 位只读。

**2.** XTSTART 与 EXCLKS 和 OSCSELS (时钟操作模式选择寄存器(OSCCTL)的第 5 位和第 4 位)结合使用。参见 **(3)　副系统时钟引脚操作模式的设置。**

**注意事项　必须将第 3 位和第 7 位清零。**

备注　**1.** $f_{XP}$:　主系统时钟振荡频率

**2.** $f_{SUB}$:　副系统时钟振荡频率

在 78K0/KC2 中执行速度最快的指令执行时间在 2 个 CPU 时钟以内。CPU 时钟($f_{CPU}$)与指令最短执行时间的关系如表 5-2 所示。

表 **5-2. CPU** 时钟与指令最短执行时间的关系

| CPU 时钟($f_{CPU}$) | 指令最短执行时间: 2/$f_{CPU}$ | | | |
|---|---|---|---|---|
| | 主系统时钟 | | | 副系统时钟 |
| | 高速系统时钟[注] | | 内部高速振荡时钟[注] | |
| | 操作频率：10 MHz | 操作频率：20 MHz | 操作频率： 8 MHz (TYP.) | 操作频率：32.768 kHz |
| $f_{XP}$ | 0.2 μs | 0.1 μs | 0.25 μs (TYP.) | – |
| $f_{XP}/2$ | 0.4 μs | 0.2 μs | 0.5 μs (TYP.) | – |
| $f_{XP}/2^2$ | 0.8 μs | 0.4 μs | 1.0 μs (TYP.) | – |
| $f_{XP}/2^3$ | 1.6 μs | 0.8 μs | 2.0 μs (TYP.) | – |
| $f_{XP}/2^4$ | 3.2 μs | 1.6 μs | 4.0 μs (TYP.) | – |
| $f_{SUB}/2$ | – | | – | 122.1 μs |

注　主时钟模式寄存器(MCM)用于设置提供给 CPU 的主系统时钟(高速系统时钟/内部高速振荡时钟) (参见图 **5-6**)。

**(3)** 副系统时钟引脚操作模式的设置

使用处理器时钟控制寄存器(PCC)的第 6 位(XTSTART)和时钟操作模式选择寄存器(OSCCTL)的第 5 位与第 4 位(EXCLKS，OSCSELS)共同设置副系统时钟引脚操作模式。

表 **5-3.** 副系统时钟引脚操作模式的设置

| PCC | OSCCTL | | 副系统时钟引脚操作模式 | P123/XT1 引脚 | P124/XT2/EXCLKS 引脚 |
|---|---|---|---|---|---|
| 第 6 位 | 第 5 位 | 第 4 位 | | | |
| XTSTART | EXCLKS | OSCSELS | | | |
| 0 | 0 | 0 | I/O 端口模式 | I/O 端口 | |
| 0 | 0 | 1 | XT1 振荡模式 | 晶体振荡器连接 | |
| 0 | 1 | 0 | I/O 端口模式 | I/O 端口 | |
| 0 | 1 | 1 | 外部时钟输入模式 | I/O 端口 | 外部时钟输入 |
| 1 | × | × | XT1 振荡模式 | 晶体振荡器连接 | |

注意事项　**改变 XTSTART、EXCLKS 和 OSCSELS 的当前值时，应确保处理器时钟控制寄存器(PCC)的第 5 位(CLS) =0(CPU 使用主系统时钟)。**

备注　×：不必考虑

**(4) 内部振荡模式寄存器 (RCM)**

内部振荡模式寄存器用于设置内部振荡器操作模式。

由 1 位或 8 位存储器操作指令设置 RCM。

复位信号的产生可将该寄存器设置为 80H[注 1]。

**图 5-4. 内部振荡模式寄存器(RCM)的格式**

地址: FFA0H 复位后: 80H[注 1] R/W[注 2]

| 符号 | <7> | 6 | 5 | 4 | 3 | 2 | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| RCM | RSTS | 0 | 0 | 0 | 0 | 0 | LSRSTOP | RSTOP |

| RSTS | 内部高速振荡器状态 |
|---|---|
| 0 | 等待内部高速振荡器精确稳定 |
| 1 | 内部高速振荡器的稳定操作 |

| LSRSTOP | 内部低速振荡器振荡/停止 |
|---|---|
| 0 | 内部低速振荡器振荡 |
| 1 | 内部低速振荡器停止 |

| RSTOP | 内部高速振荡器振荡/停止 |
|---|---|
| 0 | 内部高速振荡器振荡 |
| 1 | 内部高速振荡器停止 |

注 **1.** 复位释放后寄存器的值立即变为 00H，但在内部高速振荡器稳定后又自动变为 80H。
**2.** 第 7 位为只读。

**注意事项 当设置 RSTOP =1 时，必须确保 CPU 使用的不是内部高速振荡时钟。特别地，在下列任一条件下，可以设置 RSTOP =1。**

- **当 MCS = 1 (当 CPU 使用高速系统时钟时)**
- **当 CLS = 1 (当 CPU 使用副系统时钟时)**

**此外，在将 RSTOP 设置为 1 之前应停止正在使用内部高速振荡时钟操作的外部硬件。**

**(5) 主 OSC 控制寄存器 (MOC)**

MOC 用于选择高速系统时钟的操作模式。

当 CPU 不使用高速系统时钟时，该寄存器用于停止 X1 振荡器或禁止 EXCLK 引脚的外部时钟输入。

可以用 1 位或 8 位存储器操作指令设置 MOC。

复位信号的产生可将该寄存器设置为 80H。

**图 5-5. 主 OSC 控制寄存器 (MOC)的格式**

地址: FFA2H 复位后: 80H R/W

| 符号 | <7> | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| MOC | MSTOP | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| MSTOP | 高速系统时钟操作控制 | |
|---|---|---|
| | X1 振荡模式 | 外部时钟输入模式 |
| 0 | X1 振荡器操作 | 允许使用来自 EXCLK 引脚的外部时钟 |
| 1 | X1 振荡器停止 | 禁止使用来自 EXCLK 引脚的外部时钟 |

注意事项 **1.** **设置 MSTOP = 1 时，必须确保 CPU 使用的不是高速系统时钟。特别地，在下列任一条件下，可以设置 MSTOP = 1。**

- **当 MCS = 0 (当 CPU 使用内部高速振荡时钟时)**
- **当 CLS = 1 (当 CPU 使用副系统时钟时)**

**此外，在设置 MSTOP 为 1 之前应停止正在使用高速系统时钟操作的外部硬件。**

**2.** **当时钟操作模式选择寄存器(OSCCTL)的第 6 位(OSCSEL)= 0(I/O 端口模式)时，不要将 MSTOP 清零。**

**3.** **外部硬件时钟停止时外部硬件不能操作。外部硬件时钟停止后，若要恢复外部硬件的操作，则必须初始化外部硬件。**

**(6)　主时钟模式寄存器 (MCM)**

MCM 用于选择提供给 CPU 的时钟和外部硬件时钟的主系统时钟。

由 1 位或 8 位存储器操作指令设置 MCM。

复位信号的产生可对寄存器清零(00H)。

**图 5-6. 主时钟模式寄存器 (MCM)的格式**

地址:　FFA1H　　复位后:　00H　　R/W[注]

| 符号 | 7 | 6 | 5 | 4 | 3 | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| MCM | 0 | 0 | 0 | 0 | 0 | XSEL | MCS | MCM0 |

<table>
<tr><th rowspan="2">XSEL</th><th rowspan="2">MCM0</th><th colspan="2">主系统和外部硬件的时钟选择</th></tr>
<tr><th>主系统时钟 ($f_{XP}$)</th><th>外部硬件时钟 ($f_{PRS}$)</th></tr>
<tr><td>0</td><td>0</td><td rowspan="3">内部高速振荡时钟($f_{RH}$)</td><td rowspan="2">内部高速振荡时钟 ($f_{RH}$)</td></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">高速系统时钟 ($f_{XH}$)</td></tr>
<tr><td>1</td><td>1</td><td>高速系统时钟 ($f_{XH}$)</td></tr>
</table>

| MCS | 主系统时钟状态 |
|---|---|
| 0 | 使用内部高速振荡时钟 |
| 1 | 使用高速系统时钟 |

注　　第 1 位只读。

注意事项　**1.XSEL 在复位释放后只能被修改一次。**

**2.无论 XSEL 和 MCM0 如何设置，除了 fPRS 以外的一个时钟将被提供给如下外围硬件功能**

- **看门狗定时器(使用内部低速振荡时钟)**
- **选择“$f_{RL}$”, “$f_{RL}/2^7$”, 或者 “$f_{RL}/2^9$”作为 8 位定时器H1 的计数时钟(使用内部低速振荡时钟)**
- **选择外部时钟作为外部硬件的时钟源**
  **(以下情况除外：　选择 TM0n (n = 0, 1)的外部计数时钟(TI00n 有效沿)时)**

**(7) 振荡稳定时间计数器的状态寄存器 (OSTC)**

该寄存器用于指示 X1 时钟振荡稳定时间计数器的计数状态。当 X1 时钟振荡启动时，使用内部高速振荡时钟或副系统时钟作为 CPU 时钟，X1 时钟振荡稳定时间可以被检测。

可以由 1 位或 8 位存储器操作指令读取 OSTC 的内容。

复位释放后(由 $\overline{\text{RESET}}$ 输入、POC、LVI、和 WDT 进行复位)，通过 STOP 指令，以及设置 MSTOP(MOC 寄存器的第 7 位) = 1，可以对 OSTC 清零(00H)。

**图 5-7. 振荡稳定时间计数器的状态寄存器(OSTC)的格式**

地址: FFA3H 复位后: 00H R

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| OSTC | 0 | 0 | 0 | MOST11 | MOST13 | MOST14 | MOST15 | MOST16 |

| MOST11 | MOST13 | MOST14 | MOST15 | MOST16 | 振荡稳定时间的状态 | | |
|---|---|---|---|---|---|---|---|
| | | | | | | $f_X$ = 10 MHz | $f_X$ = 20 MHz |
| 1 | 0 | 0 | 0 | 0 | $2^{11}/f_X$ min. | 204.8 μs min. | 102.4 μs min. |
| 1 | 1 | 0 | 0 | 0 | $2^{13}/f_X$ min. | 819.2 μs min. | 409.6 μs min. |
| 1 | 1 | 1 | 0 | 0 | $2^{14}/f_X$ min. | 1.64 ms min. | 819.2 μs min. |
| 1 | 1 | 1 | 1 | 0 | $2^{15}/f_X$ min. | 3.27 ms min. | 1.64 ms min. |
| 1 | 1 | 1 | 1 | 1 | $2^{16}/f_X$ min. | 6.55 ms min. | 3.27 ms min. |

注意事项

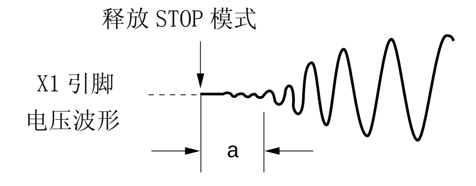
**1. 在经过上述稳定时间后，从 MOST11 开始的位依次被设置为 1，并一直保持。**

**2. 振荡稳定时间计数器值增加到由 OSTS 设置的振荡稳定时间。当内部高速振荡时钟作为 CPU 时钟，如果已进入 STOP 模式，在释放该模式时，按如下方式设置振荡稳定时间。**

- **预期的 OSTC 振荡稳定时间 ≤ 由 OSTS 设置的振荡稳定时间。**

**注意，在 STOP 模式释放后，仅将达到振荡稳定时间(由 OSTS 设置)的状态赋给 OSTC。**

**3. X1 时钟振荡稳定等待时间不包括时钟振荡开始之前的时间(下图“a”表示的部分)。**




备注 $f_X$: X1 时钟振荡频率

**(8) 振荡稳定时间选择寄存器 (OSTS)**

OSTS 用于选择 X1 时钟振荡稳定等待时间(当 STOP 模式释放时)。

当 CPU 使用 X1 时钟且 STOP 模式释放后，等待由 OSTS 设置的时间。

当 CPU 使用内部高速振荡时钟时，应使用 OSTC 确认 STOP 模式释放后已经经历了所要求的振荡稳定时间。可以使用 OSTC 检测振荡稳定时间。

可以由 8 位存储器操作指令设置 OSTS。

复位信号的产生将 OSTS 内容设置为 05H。

**图 5-8. 振荡稳定时间选择寄存器(OSTS)的格式**

地址: FFA4H　复位后: 05H　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| OSTS | 0 | 0 | 0 | 0 | 0 | OSTS2 | OSTS1 | OSTS0 |

| OSTS2 | OSTS1 | OSTS0 | 振荡稳定时间的选择 | | |
|---|---|---|---|---|---|
| | | | | $f_X$ = 10 MHz | $f_X$ = 20 MHz |
| 0 | 0 | 1 | $2^{11}/f_X$ | 204.8 μs | 102.4 μs |
| 0 | 1 | 0 | $2^{13}/f_X$ | 819.2 μs | 409.6 μs |
| 0 | 1 | 1 | $2^{14}/f_X$ | 1.64 ms | 819.2 μs |
| 1 | 0 | 0 | $2^{15}/f_X$ | 3.27 ms | 1.64 ms |
| 1 | 0 | 1 | $2^{16}/f_X$ | 6.55 ms | 3.27 ms |
| 其它情况 | | | 禁止设置 | | |

注意事项

**1.** 当 **CPU** 使用 **X1** 时钟时，若要设置 **STOP** 模式，则必须在执行 **STOP** 指令之前设置 **OSTS**。

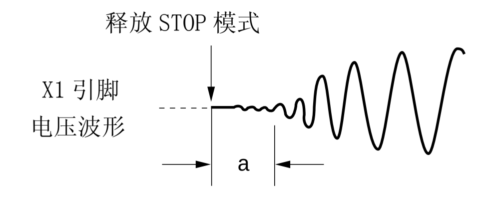
**2.** 在 **X1** 时钟振荡稳定时间内不要改变 **OSTS** 的值。

**3.** 振荡稳定时间计数器值增加到由 **OSTS** 设置的振荡稳定时间。**CPU** 使用内部高速振荡时钟时，如果已进入 **STOP** 模式，在释放该模式时，按如下方式设置振荡稳定时间。

- 预期的 **OSTC** 振荡稳定时间 ≤ 由 **OSTS** 设置的振荡稳定时间。

注意，在 **STOP** 模式释放后，仅将达到振荡稳定时间的状态**(**由 **OSTS** 设置**)**赋给 **OSTC**。

**4. X1** 时钟振荡稳定等待时间不包括时钟振荡开始之前的时间**(**下图**“a”**表示的部分**)**。




备注　$f_X$: X1 时钟振荡频率

## 5.4 系统时钟振荡器

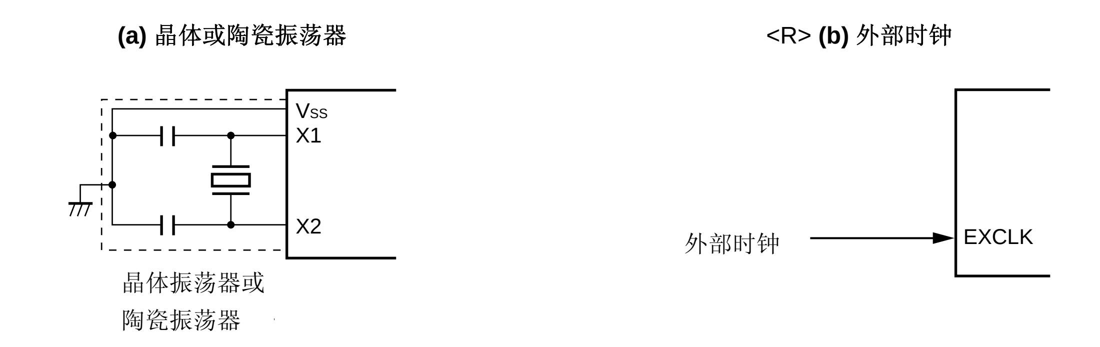
### 5.4.1　X1 振荡器

X1 振荡器采用晶体振荡器或陶瓷振荡器(1 到 20MHz，连接到 X1 和 X2 引脚)进行操作。

<R> 可以输入一个外部时钟。在本例中，输入了一个时钟信号到 EXCLK 引脚。

图 5-9 为 X1 振荡器的外部电路示例。

**图 5-9. X1 振荡器的外部电路示例**




使用时的注意事项在下页列出。

### 5.4.2 XT1 振荡器

XT1 振荡器采用晶体振荡器(连接到 XT1 和 XT2 引脚，标准值为: 32.768 kHz)进行操作。

<R> 可以输入一个外部时钟。在本例中，输入了一个时钟信号到 EXCLK 引脚。

图 5-10 为 XT1 振荡器的外部电路示例。

**图 5-10. XT1 振荡器的外部电路示例**




使用时的注意事项在下页列出。

注意事项　**1.** 在使用 **X1** 振荡器和 **XT1** 振荡器时，图 **5-9** 和图 **5-10** 中被虚线包围的部分的配线应按照如下配线方法配线，以防止连接线电容产生不利影响。

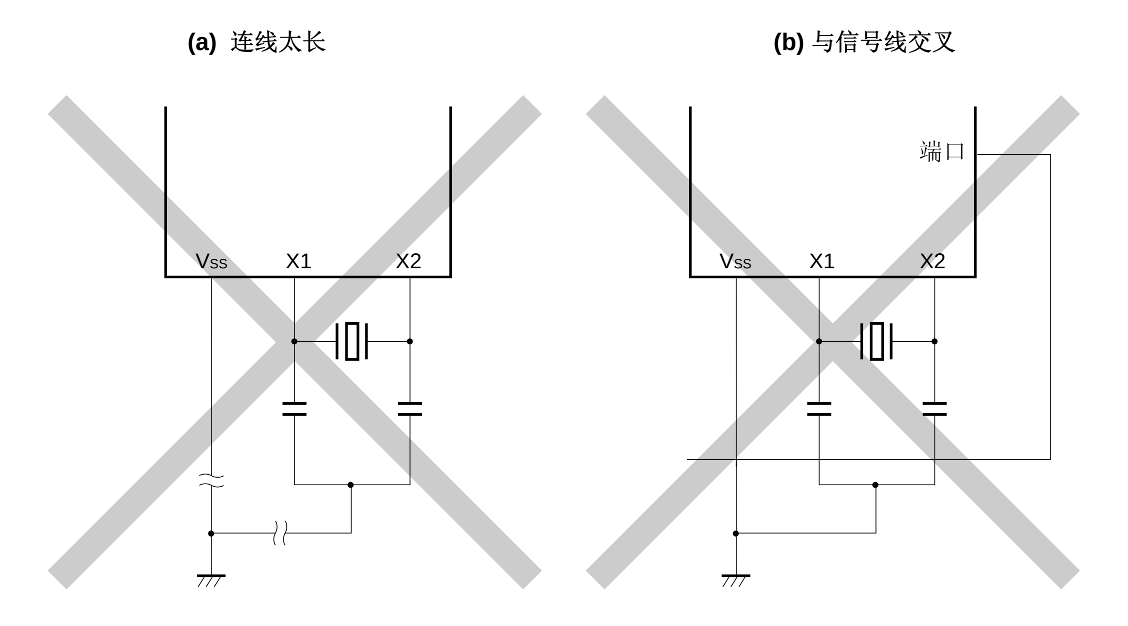
- 连接线越短越好。
- 连接线不应与其他信号线交叉。流经的电流变化较大的信号线不要在振荡器周围布线。
- 要保持振荡器电容器的接地点电压与 $V_{SS}$ 相同。不要将电容的地信号接入大电流地。
- 不要从振荡器获取信号。

注意，**XT1** 振荡器被设计成低振幅电路，以降低功耗。

图 5-11 为不正确的振荡器连接示例。

图 **5-11.** 不正确的振荡器连接示例 **(1/2)**

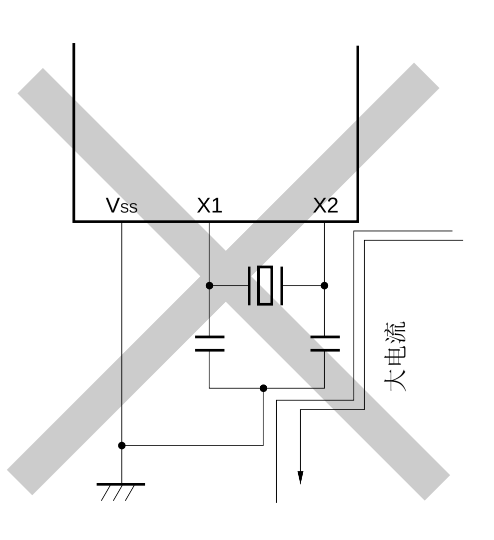
**(a)** 连线太长　　**(b)** 与信号线交叉




备注　在使用副系统时钟时，分别用 XT1 和 XT2 代替 X1 和 X2。串联电阻也插在 XT2 这边。

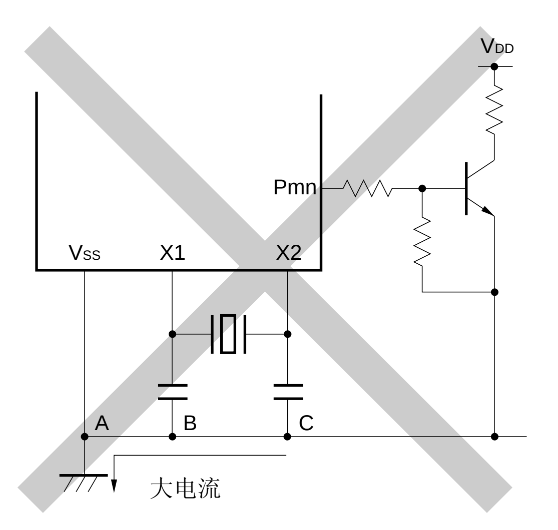
图 **5-11.** 不正确的振荡器连接示例 **(2/2)**

**(c)** 连接线周围有高变化电流

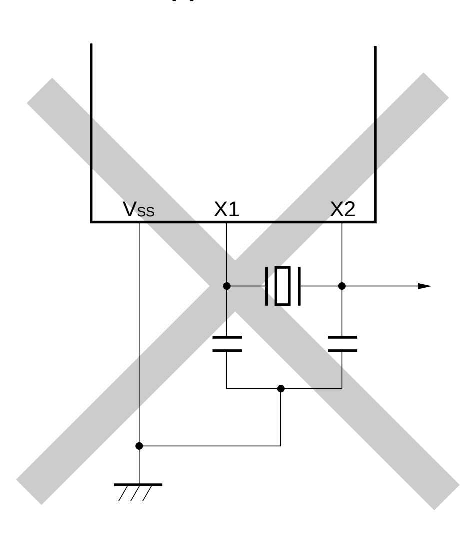



**(d)** 大电流经过振荡器的地线


**(A、 B、C** 三点电势波动**)**




**(e)** 获取信号




备注　　在使用副系统时钟时，分别用 XT1 和 XT2 代替 X1 和 X2。串联电阻也插在 XT2 这边。

注意事项　**2.**当 **X2** 和 **XT1** 并行连接时，**X2** 的串扰噪音会叠加到 **XT1**，从而产生错误。

### 5.4.3 不使用副系统时钟

如果无需使用副系统时钟进行低功耗操作，或者不使用副系统时钟作为 I/O 端口，则可以设置 XT1 和 XT2 引脚为 I/O 模式(OSCSELS = 0)，并按下列方式进行连接。

输入 (PM123/PM124 = 1):　通过一个电阻独立连接到 $V_{DD}$ 或 $V_{SS}$。
输出 (PM123/PM124 = 0):　保持开路。

备注　OSCSELS:　时钟操作模式选择寄存器的第 4 位 (OSCCTL)
PM123, PM124:　端口模式寄存器 12 的第 3 位和第 4 位 4 (PM12)

### 5.4.4 内部高速振荡器

78K0/KC2 产品中包含内部高速振荡器。可以通过内部振荡模式寄存器(RCM)控制振荡。

复位释放后，内部高速振荡器自动开始振荡(8 MHz (TYP.))。

### 5.4.5 内部低速振荡器

78K0/KC2 产品中包含内部低速振荡器。

内部低速振荡时钟只作为看门狗定时器和 8 位定时器 H1 的时钟使用。内部低速振荡时钟不能用作 CPU 时钟。

可以通过选项字节选择内部低速振荡器“可由软件停止”或“不能停止”。如果设置“可由软件停止”，则可由内部振荡模式寄存器(RCM)控制振荡。

复位释放后，内部低速振荡器自动产生振荡，同时如果使用选项字节允许看门狗定时器操作，则可以驱动看门狗定时器 (240 kHz (TYP.))。

### 5.4.6 预分频器

当 CPU 使用主系统时钟时，通过分频主系统时钟，预分频器可以产生多种时钟。

## 5.5 时钟发生器的操作

时钟发生器用于产生以下几种时钟，并控制 CPU 的操作模式，如待机模式。(见图 **5-1**)。

- 主系统时钟 $f_{XP}$
  - 高速系统时钟 $f_{XH}$
    - X1 时钟　$f_X$
    - 外部主系统时钟　$f_{EXCLK}$
  - 内部高速振荡时钟　$f_{RH}$
- 副系统时钟　$f_{SUB}$
  - XT1 时钟　$f_{XT}$
  - 外部副系统时钟　$f_{EXCLKS}$
- 内部低速振荡时钟　$f_{RL}$
- CPU 时钟　$f_{CPU}$
- 外部硬件时钟　$f_{PRS}$

在 78K0/KF2 中，当复位释放后内部高速振荡器输出时，CPU 开始操作，因此具有以下特点。

**(1)** 增强安全功能

如果 X1 时钟被默认设置为 CPU 时钟，在 X1 时钟遭到损坏或连接错误时设备不能操作，因此复位释放后也不能操作。但是，如果 CPU 的初始时钟是内部高速振荡时钟，则在复位释放后，由内部高速振荡时钟启动设备。这样，系统只需执行最少操作(如由软件确认复位源或在出现故障时执行安全处理)，便可以安全关闭。

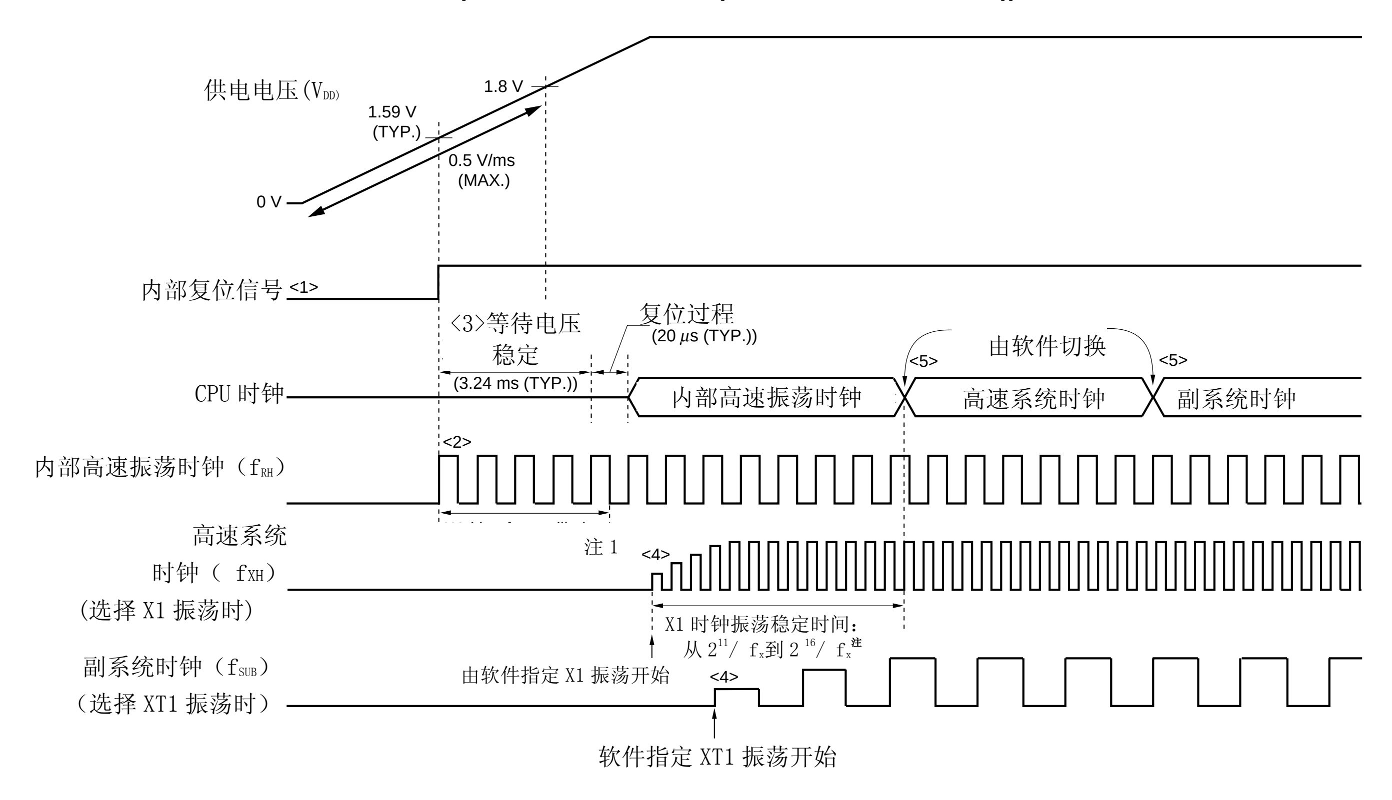
**(2)** 改善性能

由于 CPU 可以在不必等待 X1 时钟振荡稳定时间情况下就启动，所以总的性能得到了改善。

图 5-12 为上电时，时钟发生器的操作图。

<R>

**图 5-12. 上电时的时钟发生器操作**
**(设置 1.59 V POC 模式 (选项字节: POCMODE = 0))**




<1> 当电源开启时，通过上电清零 (POC) 电路产生一个内部复位信号。

<2> 当供电电压超过 1.59 V (TYP.)，复位释放并且内部高速振荡器自动开始振荡。

<R> <3> 当供电电压上升斜率为 0.5 V/ms (MAX.)时，复位释放并且在经历了供电电压和稳压器的稳定时间后，CPU 开始使用内部高速振荡时钟，然后进行复位处理。

<4> 通过软件设置 X1 或 XT1 时钟的振荡开始(参见 **5.6.1 高速系统时钟控制示例**中的**(1)**和 **5.6.3 副系统时钟控制示例**中的**(1)**)。

<5> 当 CPU 时钟切换到 X1 或 XT1 时钟时，等待时钟振荡稳定，然后通过软件设置切换(参见 **5.6.1 高速系统时钟控制示例**中的**(3)**和 **5.6.3 副系统时钟控制示例**中的**(3)**)。

<R> 注 **1.内部电压稳定时间包括内部高速振荡时钟的振荡精确稳定时间。**

**2.**释放复位后(如上图所示)或在 CPU 使用内部高速振荡时钟时释放 STOP 模式后，使用振荡稳定时间计数器的状态寄存器(OSTC)来确认 X1 时钟的振荡稳定时间。如果 CPU 使用高速系统时钟(X1 振荡)，则可使用振荡稳定时间选择寄存器(OSTS)设置释放 STOP 模式时的振荡稳定时间。

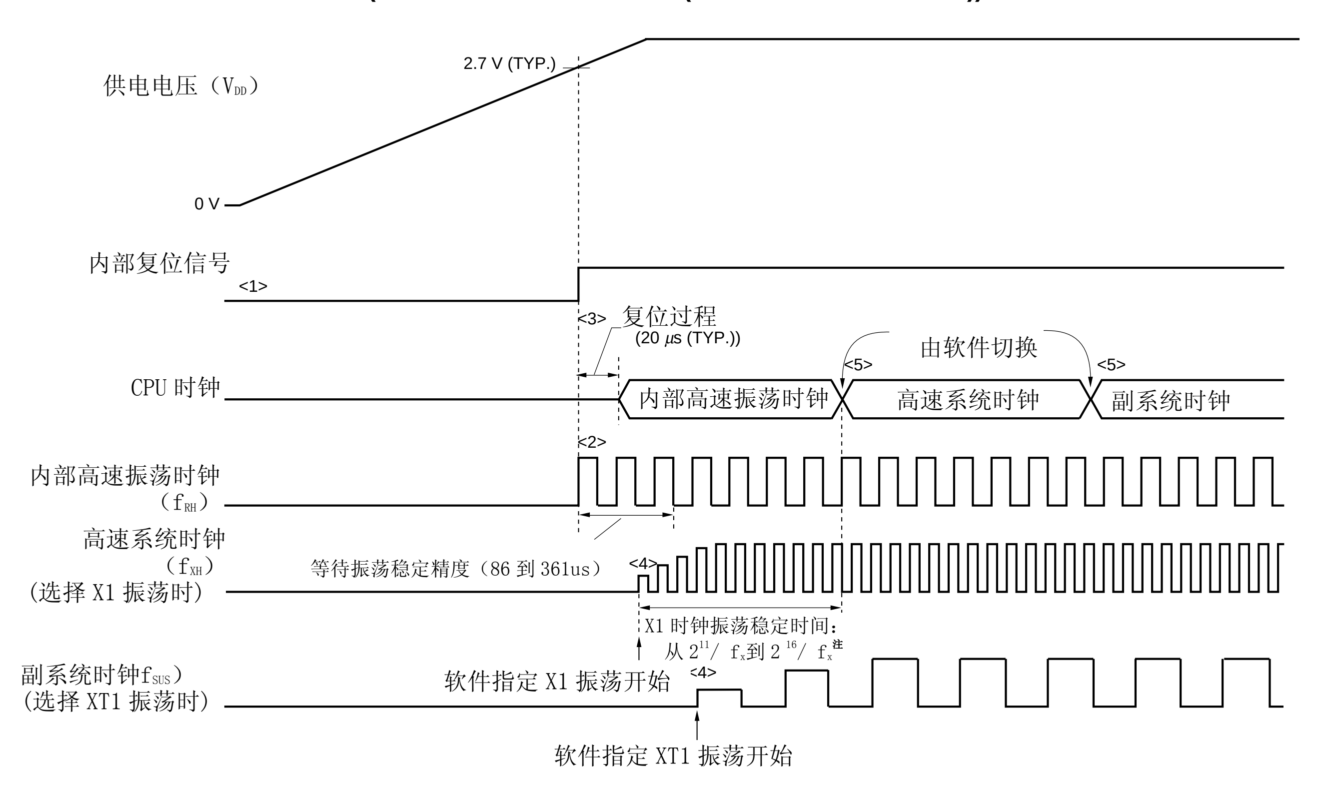
<R>　注意事项　1. 在供电电压达到 **1.8V** 之前，如果电压上升斜率小于 **0.5 V/ms (MAX.)**，则输入一个低电平到电源的 **RESET** 引脚直至电压达到 **1.8V**，或者通过使用选项字节**(POCMODE = 1)**设置 **2.7 V/1.59 V POC** 模式(见图 **5-13**) 。通过这种方式，**CPU** 的操作时序与 **<2>** 及由 **RESET** 引脚释放复位后的时序相同，如图 **5-12** 所示 。

2. 当使用 **EXCLK** 和 **EXCLKS** 引脚的外部时钟输入时，不需要等待振荡稳定时间。

备注　当微控制器正在操作时，不作为 CPU 时钟使用的时钟可以通过软件设置来停止。内部高速振荡时钟和高速系统时钟可以通过执行 STOP 指令来停止(参见 **5.6.1 高速系统时钟控制示例**中的**(4)**和 **5.6.3 副系统时钟控制示例**中的**(4)**)。

**图 5-13　上电时的时钟发生器操作**

**(设置 2.7 V/1.59 V POC 模式 (选项字节: POCMODE = 1))**




<1>　当电源开启时，通过上电清零 (POC) 电路产生一个内部复位信号。

<2>　当供电电压超过 2.7 V (TYP.)，复位释放并且内部高速振荡器自动开始振荡。

<3>　复位释放并进行了复位处理后，CPU 开始使用内部高速振荡时钟操作。

<4>　通过软件设置 X1 或 XT1 时钟的振荡开始(参见 **5.6.1 高速系统时钟控制示例**中的**(1)**和 **5.6.3 副系统时钟控制示例**中的**(1)**)。

<5>　当 CPU 时钟切换到 X1 或 XT1 时钟时，等待时钟振荡稳定，然后通过软件设置切换(参见 **5.6.1 高速系统时钟控制示例**中的**(3)**和 **5.6.3 副系统时钟控制示例**中的**(3)**)。

注　释放复位后(如上图所示)或在 CPU 使用内部高速振荡时钟时释放 STOP 模式后，使用振荡稳定时间计数器的状态寄存器(OSTC)来确认 X1 时钟的振荡稳定时间。如果 CPU 使用高速系统时钟(X1 振荡)，则可使用振荡稳定时间选择寄存器(OSTS)设置释放 STOP 模式时的振荡稳定时间。

<R>　注意事项　**1. 当电源电压达到 1.59 V (TYP.) 时，需要一个 1.93 到 5.39 ms 的振荡稳定时间。如果在 1.93 ms 以内，电源电压从 1.59 V (TYP.) 升到 2.7 V (TYP.) , 则在复位处理以前，自动产生一个 0 到 5.39 ms 的振荡稳定时间。**

**2. 当使用 EXCLK 和 EXCLKS 引脚的外部时钟输入时，不需要等待振荡稳定时间。**

**备注**　当微控制器正在操作时，不作为 CPU 时钟使用的时钟可以通过软件设置来停止。内部高速振荡时钟和高速系统时钟可以通过执行 STOP 指令来停止(参见 **5.6.1 高速系统时钟控制示例**中的**(4)**，**5.6.2 内部高速振荡时钟控制示例**中的**(3)** 和 **5.6.3 副系统时钟控制示例**中的**(4)**)。

## 5.6 时钟控制

### 5.6.1 高速系统时钟控制

如下两种高速系统时钟可用。

- X1 时钟:　　　　　　　　晶体/陶瓷振荡器连接到 X1 和 X2 引脚。
- 外部主系统时钟:　　　　外部时钟输入到 EXCLK 引脚。

不使用高速系统时钟时，X1/P121 和 X2/EXCLK/P122 引脚可用作 I/O 端口引脚。

**注意事项　复位释放后，X1/P121 和 X2/EXCLK/P122 引脚为 I/O 端口模式。**

下面描述的示例是以下情况时的设置过程示例。

(1) X1 时钟振荡

(2) 使用外部主系统时钟

(3) 将高速系统时钟作为 CPU 时钟和外部硬件时钟

(4) 停止高速系统时钟

**(1) X1 时钟振荡时设置过程示例**

<1> 设置频率(OSCCTL 寄存器)

使用 AMPH，根据使用频率设置内置振荡器。

| AMPH[注] | 操作频率控制 |
|---|---|
| 0 | 1 MHz ≤ $f_{XH}$ ≤ 10 MHz |
| 1 | 10 MHz < $f_{XH}$ ≤ 20 MHz |

<R>　**注**　复位释放后设置外部功能前设置 AMPH。复位释放后 AMPH 的值只能被修改一次。当 AMPH＝1 时，CPU 时钟停止 4.06 到 16.12 μs。

**备注**　$f_{XH}$: 高速系统时钟振荡频率

<2> 设置 P121/X1 和 P122/X2/EXCLK 引脚，并选择 X1 时钟或外部时钟(OSCCTL 寄存器)，

当 EXCLK 清零且 OSCSEL 置 1 时，从端口模式切换到 X1 振荡模式

| EXCLK | OSCSEL | 高速系统时钟引脚操作模式 | P121/X1 引脚 | P122/X2/EXCLK 引脚 |
|---|---|---|---|---|
| 0 | 1 | X1 振荡模式 | 晶体/陶瓷振荡器连接 | |

<3> 控制 X1 时钟振荡 (MOC 寄存器)

若 MSTOP 被清零，则 X1 振荡器开始振荡。

<4> 等待 X1 时钟振荡稳定

检测 OSTC 寄存器并等待所需的时间。

在等待时间内，其它软件处理过程的执行可以使用内部高速振荡时钟。

注意事项 **1.**在 **X1** 时钟运行时，不要改变 **EXCLK** 和 **OSCSEL** 的值。
**2.**当供电电压达到所用时钟的操作电压后，设置 **X1** 时钟**(**参见 第二十八章 电子产品规范**(**标准产品**)** 和 第二十九章 电子产品规范**((A)**等级产品，目标 **))**。

**(2)** 使用外部主系统时钟时设置过程示例

<1> 设置频率(OSCCTL 寄存器)

使用 AMPH, 设置使用频率。

| AMPH[注] | 操作频率控制 |
|---|---|
| 0 | 1 MHz ≤ $f_{XH}$ ≤ 10 MHz |
| 1 | 10 MHz < $f_{XH}$ ≤ 20 MHz |

<R> 注 复位释放后设置外部功能前设置 AMPH。复位释放后 AMPH 的值只能被修改一次。当 AMPH＝1 时，在 160 个外部时钟的时间内，将停止向 CPU 提供时钟 。

备注 $f_{XH}$: 高速系统时钟振荡频率

<2> 设置 P121/X1 和 P122/X2/EXCLK 引脚并选择操作模式(OSCCTL 寄存器)

当 EXCLK 和 OSCSEL 置 1 时，从端口模式切换到外部时钟输入模式。

| EXCLK | OSCSEL | 高速系统时钟引脚操作模式 | P121/X1 引脚 | P122/X2/EXCLK 引脚 |
|---|---|---|---|---|
| 1 | 1 | 外部时钟输入模式 | I/O 端口 | 外部时钟输入 |

<3> 控制外部主系统时钟输入(MOC 寄存器)

MSTOP 清零时，允许外部主系统时钟输入。

注意事项 **1.** 在外部主系统时钟运行时，不要改变 **EXCLK** 和 **OSCSEL** 的值。
**2.** 当供电电压达到所用时钟的操作电压后，设置外部主系统时钟**(**第二十八章 电子产品规范**(**标准产品**)** 和 第二十九章 电子产品规范**((A)**等级产品，目标**))**。

**(3)** 使用高速系统时钟作为 **CPU** 时钟和外部硬件时钟时设置过程示例

<1> 设置高速系统时钟振荡[注]

(参见 **5.6.1 (1) X1** 时钟振荡时设置过程示例和 **(2)** 使用外部主系统时钟时设置过程示例。)

注 当高速系统时钟已经运行时，不需要设置<1>。

<2>　设置高速系统时钟作为主系统时钟(MCM 寄存器)

XSEL 和 MCM0 置 1 时，高速系统时钟作为主系统时钟和外部硬件时钟 。

| XSEL | MCM0 | 主系统时钟和外部硬件时钟的选择 | |
|---|---|---|---|
| | | 主系统时钟 ($f_{XP}$) | 外部硬件时钟 ($f_{PRS}$) |
| 1 | 1 | 高速系统时钟 ($f_{XH}$) | 高速系统时钟 ($f_{XH}$) |

**注意事项　如果选择高速系统时钟作为主系统时钟，则只有高速系统时钟可以被设置为外部硬件时钟。**

<3>　设置主系统时钟作为 CPU 时钟并选择分频比(PCC 寄存器)

CSS 清零时，主系统时钟提供给 CPU。若要选择 CPU 时钟分频比，应使用 PCC0、PCC1 和 PCC2。

| CSS | PCC2 | PCC1 | PCC0 | CPU 时钟 ($f_{CPU}$) 选择 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $f_{XP}$ |
| | 0 | 0 | 1 | $f_{XP}/2$ (默认) |
| | 0 | 1 | 0 | $f_{XP}/2^2$ |
| | 0 | 1 | 1 | $f_{XP}/2^3$ |
| | 1 | 0 | 0 | $f_{XP}/2^4$ |
| | 其它情况 | | | 禁止设置 |

**(4)　停止高速系统时钟时设置过程示例**

高速系统时钟可以用如下两种方式停止。

- 执行 STOP 指令并停止 X1 振荡（如果应用外部时钟，则停止时钟输入）
- MSTOP 置 1 并停止 X1 振荡(如果使用外部时钟则禁止时钟输入)

**(a) 执行 STOP 指令**

<1>　停止外部硬件的设置

停止那些不能在 STOP 模式下使用的外部硬件(需要了解不能在 STOP 模式下使用的外部硬件，可参见**第二十章　待机功能** )

<2>　释放待机模式后设置 X1 时钟振荡稳定时间

当 CPU 使用 X1 时钟时，在 STOP 指令执行前设置 OSTS。

<3>　执行 STOP 指令

执行 STOP 指令时，系统处于 STOP 模式下并且 X1 振荡停止 (禁止外部时钟输入)。

**(b) 通过将 MSTOP 设置为 1 来停止 X1 振荡(禁止外部时钟输入)**

<1> 确认 CPU 时钟状态(PCC 和 MCM 寄存器)

根据 CLS 和 MCS 确认 CPU 没有使用高速系统时钟。

当 CLS = 0 且 MCS = 1 时，CPU 采用高速系统时钟，因此将 CPU 时钟切换为副系统时钟或者内部高速振荡时钟。

| CLS | MCS | CPU 时钟状态 |
|---|---|---|
| 0 | 0 | 内部高速振荡时钟 |
| 0 | 1 | 高速系统时钟 |
| 1 | × | 副系统时钟 |

<2> 停止高速系统时钟(MOC 寄存器)

MSTOP 置 1 时，X1 振荡停止(禁止外部时钟输入)。

注意事项　**当 MSTOP= 1 时，必须确认 MCS = 0 或 CLS = 1。此外，必须停止正在使用高速系统时钟的外部硬件。**

### 5.6.2 控制内部高速振荡时钟示例

下面描述的示例是以下情况时的设置过程示例。

(1)内部高速振荡时钟重新开始振荡时

(2) 使用内部高速振荡时钟作为 CPU 时钟，并且内部高速振荡时钟或高速系统时钟用作外部硬件时钟时

(3) 停止内部高速振荡时钟时

**(1) 内部高速振荡时钟重新振荡时设置过程示例**[注 1]

<1> 内部高速振荡时钟重新振荡的设置(RCM 寄存器)

RSTOP 清零时，内部高速振荡时钟开始运行。

<2> 等待内部高速振荡时钟的振荡精确稳定时间(RCM 寄存器)。

一直到 RSTS = 1[注 2]。

注　**1.** 复位释放后，内部高速振荡器自动开始振荡并且内部高速振荡时钟作为 CPU 时钟使用。

**2.** 如果 CPU 时钟和外部硬件时钟不需要很高的精度，则无需等待时间。

**(2) 使用内部高速振荡时钟作为 CPU 时钟且内部高速振荡时钟或高速系统时钟作为外部硬件时钟时设置过程示例**

<1> • 内部高速振荡时钟重新振荡[注]

(参见 **5.6.2 (1) 内部高速振荡时钟重新振荡时设置过程示例**)。

• 高速系统时钟振荡[注]

(使用高速系统时钟作为外部硬件时钟时所需的设置。参见 **5.6.1 (1) X1 时钟振荡时设置过程示例**和**(2) 使用外部主系统时钟时设置过程示例。)**

注　　当内部高速振荡时钟或高速系统时钟已经运行时，不需要设置<1>。

<2>　选择主系统时钟和外部硬件时钟(MCM 寄存器)

使用 XSEL 和 MCM0 设置主系统时钟和外部硬件时钟。

| XSEL | MCM0 | 主系统时钟和外部硬件时钟的选择 | |
|---|---|---|---|
| | | 主系统时钟 ($f_{XP}$) | 外部硬件时钟 ($f_{PRS}$) |
| 0 | 0 | 内部高速振荡时钟 ($f_{RH}$) | 内部高速振荡时钟 ($f_{RH}$) |
| 0 | 1 | | |
| 1 | 0 | | 高速系统时钟($f_{XH}$) |

<3>　选择 CPU 时钟分频比(PCC 寄存器)

CSS 清零时，主系统时钟提供给 CPU。若要选择 CPU 时钟分频比，应使用 PCC0、PCC1 和 PCC2。

| CSS | PCC2 | PCC1 | PCC0 | CPU 时钟 ($f_{CPU}$) 选择 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $f_{XP}$ |
| | 0 | 0 | 1 | $f_{XP}/2$ (默认) |
| | 0 | 1 | 0 | $f_{XP}/2^2$ |
| | 0 | 1 | 1 | $f_{XP}/2^3$ |
| | 1 | 0 | 0 | $f_{XP}/2^4$ |
| | 其它情况 | | | 禁止设置 |

**(3)　停止内部高速振荡时钟时设置过程示例**

可以用如下两种方式停止内部高速振荡时钟。

- 执行 STOP 指令设置 STOP 模式
- 将 RSTOP 置 1 且停止内部高速振荡时钟

**(a)　执行 STOP 指令**

<1>　设置外部硬件

停止那些不能在 STOP 模式下使用的外部硬件(如需了解不能在 STOP 模式下使用的外部硬件，可参见**第二十二章 待机功能**)。

<2>　释放待机模式后 X1 时钟振荡稳定时间的设置

CPU 使用 X1 时钟时，在 STOP 指令执行前设置 OSTS。

<3>　执行 STOP 指令

执行 STOP 指令时，系统处于 STOP 模式下并且停止内部高速振荡时钟。

**(b) 通过 RSTOP 置 1 停止内部高速振荡时钟**

<1> 确认 CPU 时钟状态(PCC 和 MCM 寄存器)

根据 CLS 和 MCS 确认 CPU 没有使用内部高速振荡时钟。

CLS = 0 和 MCS = 0 时，CPU 采用内部高速振荡时钟，因此将 CPU 时钟切换为高速系统时钟或副系统时钟。

| CLS | MCS | CPU 时钟状态 |
|---|---|---|
| 0 | 0 | 内部高速振荡时钟 |
| 0 | 1 | 高速系统时钟 |
| 1 | × | 副系统时钟 |

<2> 停止内部高速振荡时钟(RCM 寄存器)

RSTOP 置 1 时，停止内部高速振荡时钟。

注意事项 **当 RSTOP= 1 时，必须确认 MCS = 1 或 CLS = 1。此外，必须停止正在使用内部高速振荡时钟的外部硬件。**

### 5.6.3 控制副系统时钟示例

可用如下两种副系统时钟。

- XT1 时钟:　　晶体/陶瓷振荡器连接到 XT1 和 XT2 引脚。
- 外部副系统时钟:　　外部时钟输入到 EXCLKS 引脚。

不使用副系统时钟时， XT1/P123 和 XT2/EXCLKS/P124 引脚可用作 I/O 端口引脚。

注意事项　**复位释放后 XT1/P123 和 XT2/EXCLKS/P124 引脚工作在 I/O 端口模式下。**

下面描述的示例是以下情况时的设置过程示例。

(1) XT1 时钟振荡

(2) 使用外部副系统时钟

(3) 使用副系统时钟作为 CPU 时钟

(4) 停止副系统时钟

**(1) XT1 时钟振荡时设置过程示例**

<1> 设置 XT1 和 XT2 引脚并选择操作模式 (PCC 和 OSCCTL 寄存器)

当 XTSTART、EXCLKS 和 OSCSELS 设置为如下模式时，将从端口模式切换到 XT1 振荡模式 。

| XTSTART | EXCLKS | OSCSELS | 副系统时钟引脚操作模式 | P123/XT1 引脚 | P124/XT2/ EXCLKS 引脚 |
|---|---|---|---|---|---|
| 0 | 0 | 1 | XT1 振荡模式 | 连接晶体/陶瓷振荡器 | |
| 1 | × | × | | | |

备注　　×: 不必考虑

<2> 等待副系统时钟振荡稳定

使用定时器功能，通过软件等待副系统时钟振荡稳定时间。

注意事项　**在副系统时钟运行时，不要改变 XTSTART, EXCLKS 和 OSCSELS 的值。**

**(2)** 使用外部副系统时钟时设置过程示例

<1> 设置 XT1 和 XT2 引脚，选择 XT1 时钟/外部时钟和控制振荡 (PCC 和 OSCCTL 寄存器)

当 XTSTART 清零并且 EXCLKS 和 OSCSELS 置 1 时，将从端口模式切换到外部时钟输入模式。在这种情况下，将外部时钟输入到 EXCLKS/XT2/P124 引脚。

| XTSTART | EXCLKS | OSCSELS | 副系统时钟引脚操作模式 | P123/XT1 引脚 | P124/XT2/EXCLKS 引脚 |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 外部时钟输入模式 | I/O 端口 | 外部时钟输入 |

注意事项 **在副系统时钟运行时，不要改变 XTSTART, EXCLKS 和 OSCSELS 的值。**

**(3)** 使用副系统时钟作为 **CPU** 时钟时设置过程示例

<1> 设置副系统时钟振荡[注]

(参见 **5.6.3 (1) XT1** 时钟振荡时设置过程示例和 **(2)** 使用外部副系统时钟时设置过程示例。)

注 当副系统时钟已经运行时，不需要设置<1>。

<2> 切换 CPU 时钟(PCC 寄存器)

CSS 置 1 时，副系统时钟提供给 CPU。

| CSS | PCC2 | PCC1 | PCC0 | CPU 时钟 ($f_{CPU}$) 选择 |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | $f_{SUB}/2$ |
| | 0 | 0 | 1 | |
| | 0 | 1 | 0 | |
| | 0 | 1 | 1 | |
| | 1 | 0 | 0 | |
| | 其它情况 | | | 禁止设置 |

**(4)** 停止副系统时钟时设置过程示例

<1> 确认 CPU 时钟状态(PCC 和 MCM 寄存器)

根据 CLS 和 MCS 确认 CPU 没有使用副系统时钟 。

当 CLS = 1 时，CPU 采用副系统时钟，因此将 CPU 时钟切换为内部高速振荡时钟或高速系统时钟。

| CLS | MCS | CPU 时钟状态 |
|---|---|---|
| 0 | 0 | 内部高速振荡时钟 |
| 0 | 1 | 高速系统时钟 |
| 1 | × | 副系统时钟 |

<2> 停止副系统时钟 (OSCCTL 寄存器)

OSCSELS 清零时，停止 XT1 振荡(禁止外部时钟输入)。

注意事项 **1.** **清零 OSCSEL 时，必须确认 CLS = 0。此外，如果钟表定时器使用副系统时钟，则应停止其操作。**

**2.** **副系统时钟振荡不能通过使用 STOP 指令来停止。**

### 5.6.4 控制内部低速振荡时钟示例

内部低速振荡时钟不能用作 CPU 时钟。

只有如下外部硬件可以使用这个时钟。

- 看门狗定时器
- 8 位定时器 H1(选择 $f_{RL}$ 作为计数时钟时)

此外，可以通过选项字节选择如下操作模式。

- 内部低速振荡器不能停止
- 内部低速振荡器可由软件停止

在复位释放后内部低速振荡器自动开始振荡，并且如果通过选项字节允许看门狗定时器操作，则驱动看门狗定时器 (240 kHz (TYP.))。

**(1) 停止内部低速振荡时钟时设置过程示例**

<1> 将 LSRSTOP 置 1 (RCM 寄存器)

LSRSTOP 置 1 时，停止内部低速振荡时钟。

**(2) 内部低速振荡时钟重新振荡时设置过程示例**

<1> LSRSTOP 清零 (RCM 寄存器)

LSRSTOP 清零时，内部低速振荡时钟重新振荡。

**注意事项　如果通过选项字节选择“不能停止内部低速振荡器”，则不能控制内部低速振荡时钟的振荡。**

### 5.6.5 CPU 和外部硬件所采用的时钟

下表显示了 CPU 和外部硬件采用的时钟之间的关系及寄存器的设置。

**表 5-4. CPU 和外部硬件采用的时钟及寄存器设置**

| 时钟 | | XSEL | CSS | MCM0 | EXCLK |
|---|---|---|---|---|---|
| CPU 时钟 | 外部硬件时钟 | | | | |
| 内部高速振荡时钟 | | 0 | 0 | × | × |
| 内部高速振荡时钟 | X1 时钟 | 1 | 0 | 0 | 0 |
| | 外部主系统时钟 | 1 | 0 | 0 | 1 |
| X1 时钟 | | 1 | 0 | 1 | 0 |
| 外部主系统时钟 | | 1 | 0 | 1 | 1 |
| 副系统时钟 | 内部高速振荡时钟 | 0 | 1 | × | × |
| | X1 时钟 | 1 | 1 | 0 | 0 |
| | | 1 | 1 | 1 | 0 |
| | 外部主系统时钟 | 1 | 1 | 0 | 1 |
| | | 1 | 1 | 1 | 1 |

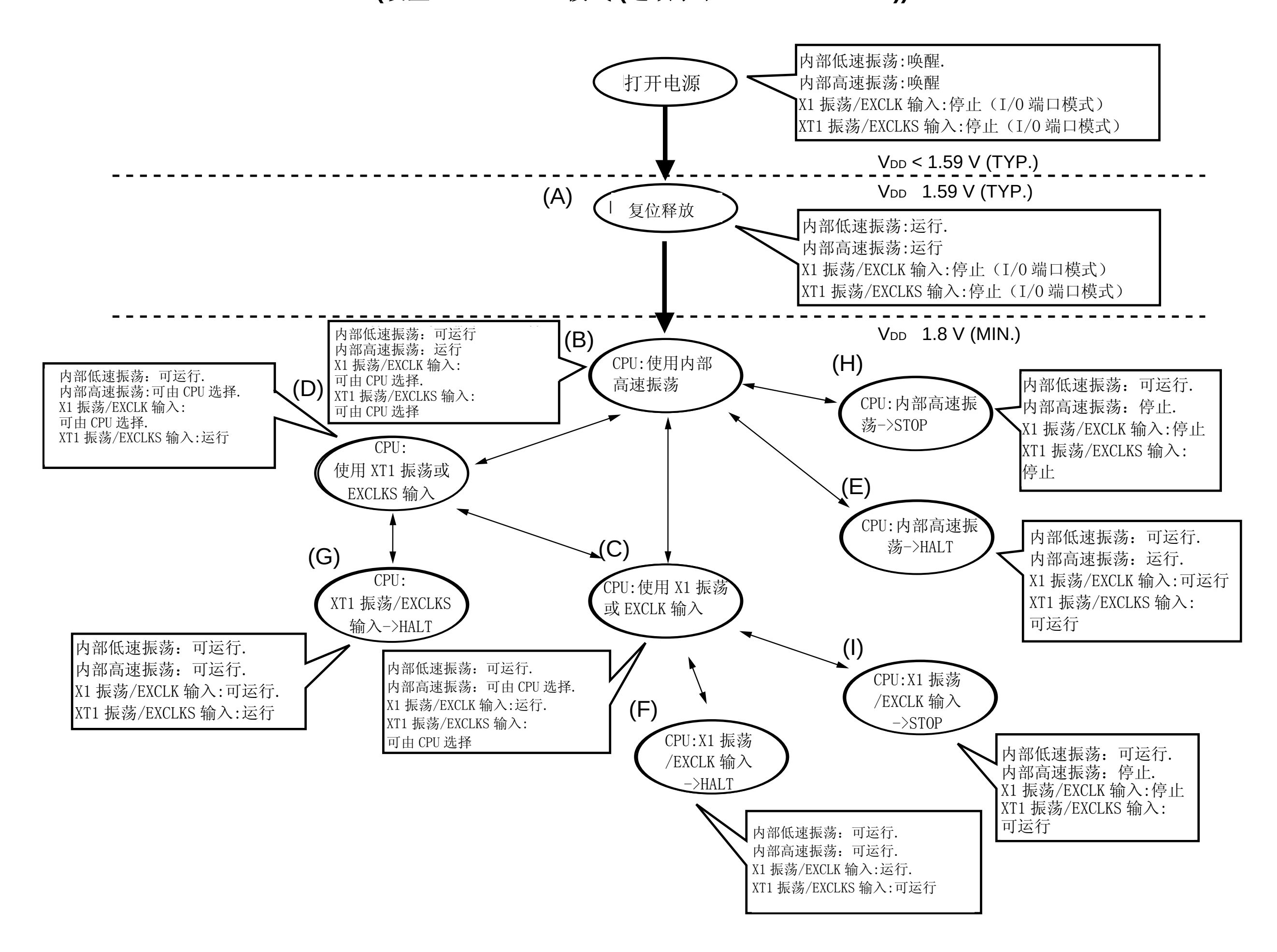
备注
1. XSEL: 主时钟模式寄存器(MCM)的第 2 位
2. CSS: 处理器时钟控制寄存器(PCC)的第 4 位
3. MCM0: MCM 的第 0 位
4. EXCLK: 时钟操作模式选择寄存器(OSCCTL)的第 7 位
5. ×: 不必考虑

### 5.6.6 CPU 时钟状态转换图

图 5-14 显示了该产品 CPU 时钟状态转换图。

**图 5-14. CPU 时钟状态转换图**

**(设置 1.59 V POC 模式 (选项字节: POCMODE = 0))**




<R>　**备注**　在 2.7 V/1.59 V POC 模式下 (选项字节: POCMODE = 1)，当供电电压超过 2.7V(TYP.)时 CPU 时钟状态转换为上图所示的(A)，而在复位处理后(11 到 45 μs )变到(B)。

表 5-5 显示了 CPU 时钟的切换过程与 SFR 寄存器设置示例

**表 5-5. CPU 时钟切换与 SFR 寄存器设置示例(1/4)**

**(1) 复位释放后(A)CPU 使用内部高速振荡时钟(B)**

| 状态转换 | SFR 寄存器设置 |
|---|---|
| (A) → (B) | SFR 寄存器不必设置 (复位释放后默认状态). |

**(2) 复位释放后(A)CPU 使用高速系统时钟(C)**
(复位释放后 CPU 立即使用内部高速振荡时钟(B)。)

( SFR 寄存器设置序列) →

| SFR 寄存器设置标志 / 状态转换 | AMPH | EXCLK | OSCSEL | MSTOP | OSTC 寄存器 | XSEL | MCM0 |
|---|---|---|---|---|---|---|---|
| (A) → (B) → (C) (X1 时钟: 1 MHz ≤ $f_{XH}$ ≤ 10 MHz) | 0 | 0 | 1 | 0 | 必须检测 | 1 | 1 |
| (A) → (B) → (C) (外部主时钟: 1 MHz ≤ $f_{XH}$ ≤ 10 MHz) | 0 | 1 | 1 | 0 | 不必检测 | 1 | 1 |
| (A) → (B) → (C) (X1 时钟: 10 MHz < $f_{XH}$ ≤ 20 MHz) | 1 | 0 | 1 | 0 | 必须检测 | 1 | 1 |
| (A) → (B) → (C) 外部主时钟: 10 MHz < $f_{XH}$ ≤ 20 MHz) | 1 | 1 | 1 | 0 | 不必检测 | 1 | 1 |

**注意事项**　**供电电压达到所用时钟的操作电压后，设置时钟(参见　第二十八章　电子产品规范(标准产品) 和 第二十九章 电子产品规范((A)等级产品，目标)) 。**

**(3) 复位释放后(A)CPU 使用副系统时钟(D)**

( SFR 寄存器设置序列) →

| SFR 寄存器设置标志 / 状态转换 | XTSTART | EXCLKS | OSCSELS | 等待振荡稳定 | CSS |
|---|---|---|---|---|---|
| (A) → (B) → (D) (XT1 时钟) | 0 | 0 | 1 | 必须 | 1 |
| | 1 | × | × | | |
| (A) → (B) → (D) (外部副系统时钟) | 0 | 1 | 1 | 不必要 | 1 |

**备注**　**1.** 表 5-5 的(A) 到 (I)对应图 5-14 的(A)到(I)。

**2.** EXCLK, OSCSEL, EXCLKS, OSCSELS, AMPH:
时钟操作模式选择寄存器(OSCCTL)的第 7 位到第 4 位与第 0 位
MSTOP: 主 OSC 控制寄存器(MOC)的第 7 位
XSEL, MCM0: 主时钟模式寄存器(MCM)的第 2 位与第 0 位
XTSTART, CSS: 处理器时钟控制寄存器(PCC)的第 6 位与第 4 位
×: 不必考虑

表 **5-5. CPU** 时钟切换与 **SFR** 寄存器设置示例**(2/4)**

**(4) CPU** 时钟从内部高速振荡时钟**(B)**切换到高速系统时钟**(C)**

(SFR 寄存器的设置顺序) →

| SFR 寄存器的设置标志 / 状态转换 | AMPH[注] | EXCLK | OSCSEL | MSTOP | OSTC 寄存器 | XSEL[注] | MCM0 |
|---|---|---|---|---|---|---|---|
| (B) → (C) (X1 时钟: 1 MHz ≤ $f_{XH}$ ≤ 10 MHz) | 0 | 0 | 1 | 0 | 必须检测 | 1 | 1 |
| (B) → (C) (外部主时钟: 1 MHz ≤ $f_{XH}$ ≤ 10 MHz) | 0 | 1 | 1 | 0 | 不必检测 | 1 | 1 |
| (B) → (C) (X1 时钟: 10 MHz < $f_{XH}$ ≤ 20 MHz) | 1 | 0 | 1 | 0 | 必须检测 | 1 | 1 |
| (B) → (C) (外部主时钟: 10 MHz < $f_{XH}$ ≤ 20 MHz) | 1 | 1 | 1 | 0 | 不必检测 | 1 | 1 |

AMPH, EXCLK, OSCSEL: 如果这些寄存器已设置则不必要

MSTOP, OSTC 寄存器: 如果 CPU 使用高速系统时钟则不必要

注　复位释放后该标志只能被修改一次。如果已经设置了该项，则无需再设置。

**注意事项　供电电压达到所用时钟的操作电压后，设置时钟(参见　第二十八章　电子产品规范(标准产品) 和 第二十九章　电子产品规范((A)等级产品，目标))。**

**(5) CPU** 时钟从内部高速振荡时钟**(B)**切换到副系统时钟**(D)**

(SFR 寄存器的设置顺序) →

| SFR 寄存器的设置标志 / 状态转换 | XTSTART | EXCLKS | OSCSELS | 等待振荡稳定 | CSS |
|---|---|---|---|---|---|
| (B) → (D) (XT1 时钟) | 0 | 0 | 1 | 必须 | 1 |
| | 1 | × | × | | |
| (B) → (D) (外部副系统时钟) | 0 | 1 | 1 | 不必要 | 1 |

XTSTART, EXCLKS, OSCSELS, 等待振荡稳定: 如果 CPU 使用副系统时钟则不必要

备注　**1.** 表 5-5 的(A) 到 (I)对应图 5-14 的(A)到(I)。

**2.** EXCLK, OSCSEL, EXCLKS, OSCSELS, AMPH: 时钟操作模式选择寄存器(OSCCTL)的第 7 位到第 4 位与第 0 位

MSTOP: 主 OSC 控制寄存器(MOC)的第 7 位

XSEL, MCM0: 主时钟模式寄存器(MCM)的第 2 位与第 0 位

XTSTART, CSS: 处理器时钟控制寄存器(PCC)的第 6 位与第 4 位

×: 不必考虑

### 表 5-5.　CPU 时钟切换与 SFR 寄存器设置示例 (3/4)

**(6) CPU 时钟从高速系统时钟(C)切换到内部高速振荡时钟(B)**

(SFR 寄存器的设置顺序) →

| SFR 寄存器的设置标志 / 状态转换 | RSTOP | RSTS | MCM0 |
|---|---|---|---|
| (C) → (B) | 0 | 确认该标志为 1 | 0 |

（RSTOP、RSTS）如果 CPU 使用内部高速振荡时钟则不必要

**(7) CPU 时钟从高速系统时钟(C)切换到副系统时钟(D)**

(SFR 寄存器的设置顺序)

| SFR 寄存器的设置标志 / 状态转换 | XTSTART | EXCLKS | OSCSELS | 等待振荡稳定 | CSS |
|---|---|---|---|---|---|
| (C) → (D) (XT1 时钟) | 0 | 0 | 1 | 必须 | 1 |
| | 1 | × | × | | |
| (C) → (D) (外部副系统时钟) | 0 | 1 | 1 | 不必要 | 1 |

（XTSTART ~ 等待振荡稳定）如果 CPU 使用副系统时钟则不必要

**(8) CPU 时钟从副系统时钟(D)切换到内部高速振荡时钟(B)**

(SFR 寄存器的设置序列) →

| SFR 寄存器的设置标志 / 状态转换 | RSTOP | RSTS | MCM0 | CSS |
|---|---|---|---|---|
| (D) → (B) | 0 | 确认该标志为 1 | 0 | 0 |

（RSTOP、RSTS）如果 CPU 使用内部高速振荡时钟则不必要

（MCM0）↑ 如果 XSEL= 0 则不必要

备注　**1.** 表 5-5 的(A) 到 (I)对应图 5-14 的(A)到(I)。

**2.** MCM0:　主时钟模式寄存器(MCM)的第 0 位
EXCLKS, OSCSELS:　时钟操作模式选择寄存器(OSCCTL)的第 5 位与第 4 位
RSTS, RSTOP:　内部振荡模式寄存器(RCM)的第 7 位和第 0 位
XTSTART, CSS:　处理器时钟控制寄存器(PCC)的第 6 位与第 4 位
×:　不必考虑

表 5-5. **CPU** 时钟切换与 **SFR** 寄存器设置示例 **(4/4)**

**(9) CPU** 时钟从副系统时钟**(D)**切换到高速系统时钟**(C)**

(SFR 寄存器的设置序列) ⟶

| SFR 寄存器的设置标志 / 状态转换 | AMPH[注] | EXCLK | OSCSEL | MSTOP | OSTC 寄存器 | XSEL[注] | MCM0 | CSS |
|---|---|---|---|---|---|---|---|---|
| (D) → (C) (X1 时钟: 1 MHz ≤ $f_{XH}$ ≤ 10 MHz) | 0 | 0 | 1 | 0 | 必须检测 | 1 | 1 | 0 |
| (D) → (C) (外部主时钟: 1 MHz ≤ $f_{XH}$ ≤ 10 MHz | 0 | 1 | 1 | 0 | 不必检测 | 1 | 1 | 0 |
| (D) → (C) (X1 时钟: 10 MHz < $f_{XH}$ ≤ 20 MHz) | 1 | 0 | 1 | 0 | 必须检测 | 1 | 1 | 0 |
| (D) → (C) (外部主时钟: 10 MHz < $f_{XH}$ ≤ 20 MHz) | 1 | 1 | 1 | 0 | 不必检测 | 1 | 1 | 0 |

如果这些寄存器已设置则不必要（AMPH, EXCLK, OSCSEL）　如果 CPU 使用高速系统时钟则不必要（MSTOP, OSTC）　如果寄存器已设置则不必要（XSEL, MCM0）

注　复位释放后该标志只能被修改一次。如果已经设置了该项，则无需再设置。

注意事项　供电电压达到所用时钟的操作电压后，设置时钟(参见　第二十八章　电子产品规范(标准产品) 和 第二十九章　电子产品规范((A)等级产品，目标)) 。

**(10) • CPU** 使用内部高速振荡时钟**(B)**时 **HALT** 模式**(E)**的设置。

**• CPU** 使用高速系统时钟**(C)**时 **HALT** 模式**(F)**的设置。

**• CPU** 使用副系统时钟**(D)**时 **HALT** 模式**(G)**的设置。

| 状态转换 | 设置 |
|---|---|
| (B) → (E)<br>(C) → (F)<br>(D) → (G) | 执行 HALT 指令 |

**(11) • CPU** 使用内部高速振荡时钟**(B)**时 **STOP** 模式**(H)** 的设置。

**• CPU** 使用高速系统时钟**(C)**时 **STOP** 模式**(I)** 的设置。

(设置顺序)

| 状态转换 | 设置 | |
|---|---|---|
| (B) → (H)<br>(C) → (I) | 停止那些不能在 STOP 模式下使用的外部功能 | 执行 STOP 指令 |

备注　**1.** 表 5-5 的(A) 到 (I)对应图 5-14 的(A)到(I)。

**2.** EXCLK, OSCSEL, AMPH:　时钟操作模式选择寄存器(OSCCTL)的第 7, 6 和 0 位
MSTOP:　主 OSC 控制寄存器(MOC)的第 7 位
XSEL, MCM0:　主时钟模式寄存器(MCM)的第 2 位和第 0 位
CSS:　处理器时钟控制寄存器(PCC) 的第 4 位

### 5.6.7 CPU 时钟切换之前的状况与切换之后的处理

CPU 时钟切换之前的状况与切换之后的处理显示如下。

**表 5-6.　切换 CPU 时钟**

| | CPU 时钟 | | 切换前的状况 | 切换后的处理 |
|---|---|---|---|---|
| | 切换前 | 切换后 | | |
| <R> | 内部高速振荡时钟 | X1 时钟 | X1 振荡稳定<br>• MSTOP = 0, OSCSEL = 1, EXCLK = 0<br>• 经历了振荡稳定时间 | • 可以停止内部高速振荡器(RSTOP = 1).<br>• AMPH 被设置为 1 后 CPU 时钟停止 4.06 ～ 16.12 μs (MIN.)。 |
| <R> | | 外部主系统时钟 | 允许来自 EXCLK 引脚的外部时钟输入<br>• MSTOP = 0, OSCSEL = 1, EXCLK = 1 | • 内部高速振荡器可停止(RSTOP = 1).<br>• 在 AMPH 被设为 1 之后，提供给 CPU 的时钟会停止 160 个从 EXCLK 引脚输入的外部时钟周期 。 |
| | X1 时钟 | 内部高速振荡时钟 | 内部高速振荡器振荡<br>• RSTOP = 0 | 可以停止 X1 振荡(MSTOP = 1). |
| | 外部主系统时钟 | | | 可以禁止外部主系统时钟输入(MSTOP = 1). |
| | 内部高速振荡时钟 | XT1 时钟 | XT1 振荡稳定<br>• XTSTART = 0, EXCLKS = 0, OSCSELS = 1 或 XTSTART = 1<br>• 经历了振荡稳定时间 | 通过停止内部高速振荡器可以降低操作电流(RSTOP = 1) |
| | X1 时钟 | | | 可以停止 X1 振荡 (MSTOP = 1) |
| | 外部主系统时钟 | | | 可以禁止外部主系统时钟输入(MSTOP = 1) |
| | 内部高速振荡时钟 | 外部副系统时钟 | 允许来自 EXCLK 引脚的外部时钟输入<br>• XTSTART = 0, EXCLKS = 1, OSCSELS = 1 | 通过停止内部高速振荡器可以降低操作电流(RSTOP = 1). |
| | X1 时钟 | | | 可以停止 X1 振荡(MSTOP = 1). |
| | 外部主系统时钟 | | | 可以禁止外部主系统时钟输入(MSTOP = 1). |
| | XT1 时钟, 外部副系统时钟 | 内部高速振荡时钟 | 内部高速振荡器振荡且选择内部高速振荡时钟作为主系统时钟<br>• RSTOP = 0, MCS = 0 | 可停止 XT1 振荡或者禁止外部副系统时钟输入(OSCSELS = 0). |
| <R> | | X1 时钟 | X1 振荡稳定且选择高速系统时钟作为主系统时钟<br>• MSTOP = 0, OSCSEL = 1, EXCLK = 0<br>• 经历了振荡稳定时间<br>• MCS = 1 | • 可停止 XT1 振荡或者禁止外部副系统时钟输入(OSCSELS = 0).<br>• AMPH 被设置为 1 后 CPU 时钟停止 4.06 ～ 16.12 μs (MIN.)。 |
| <R> | | 外部主系统时钟 | 允许来自 EXCLK 引脚的外部时钟输入 并选择高速系统时钟作为主系统时钟<br>• MSTOP = 0, OSCSEL = 1, EXCLK = 1<br>• MCS = 1 | • 可停止 XT1 振荡或者禁止外部副系统时钟输入(OSCSELS = 0).<br>• 在 AMPH 被设为 1 之后，提供给 CPU 的时钟会停止 160 个从 EXCLK 引脚输入的外部时钟周期 。 |

### 5.6.8 CPU 时钟和主系统时钟切换所需的时间

通过设置处理器时钟控制器(PCC)的第 0 位到第 2 位(PCC0 到 PCC2)以及第 4 位(CSS)，可以切换 CPU 时钟(在主系统时钟和副系统时钟之间)，同时可以改变主系统时钟的分频比。

修改 PCC 后，实际的切换操作不会立即执行；使用切换前的时钟继续操作几个时钟(见 **表 5-7**)。

可以通过 PCC 寄存器的第 5 位(CLS)来确定 CPU 使用主系统时钟还是副系统时钟。

**表 5-7. CPU 时钟切换所需时间与主系统时钟周期分频因子**

<table>
<tr><td colspan="4">切换前设置值</td><td colspan="24">切换后设置值</td></tr>
<tr><td rowspan="2">CSS</td><td rowspan="2">PCC2</td><td rowspan="2">PCC1</td><td rowspan="2">PCC0</td><td>CSS</td><td>PCC2</td><td>PCC1</td><td>PCC0</td><td>CSS</td><td>PCC2</td><td>PCC1</td><td>PCC0</td><td>CSS</td><td>PCC2</td><td>PCC1</td><td>PCC0</td><td>CSS</td><td>PCC2</td><td>PCC1</td><td>PCC0</td><td>CSS</td><td>PCC2</td><td>PCC1</td><td>PCC0</td><td>CSS</td><td>PCC2</td><td>PCC1</td><td>PCC0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>×</td><td>×</td><td>×</td></tr>
<tr><td rowspan="5">0</td><td>0</td><td>0</td><td>0</td><td colspan="4"></td><td colspan="4">16 个时钟</td><td colspan="4">16 个时钟</td><td colspan="4">16 个时钟</td><td colspan="4">16 个时钟</td><td colspan="4">2f<sub>XP</sub>/f<sub>SUB</sub> 个时钟</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td colspan="4">8 个时钟</td><td colspan="4"></td><td colspan="4">8 个时钟</td><td colspan="4">8 个时钟</td><td colspan="4">8 个时钟</td><td colspan="4">f<sub>XP</sub>/f<sub>SUB</sub> 个时钟</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td colspan="4">4 个时钟</td><td colspan="4">4 个时钟</td><td colspan="4"></td><td colspan="4">4 个时钟</td><td colspan="4">4 个时钟</td><td colspan="4">f<sub>XP</sub>/2f<sub>SUB</sub> 个时钟</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td colspan="4">2 个时钟</td><td colspan="4">2 个时钟</td><td colspan="4">2 个时钟</td><td colspan="4"></td><td colspan="4">2 个时钟</td><td colspan="4">f<sub>XP</sub>/4f<sub>SUB</sub> 个时钟</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td colspan="4">1 个时钟</td><td colspan="4">1 个时钟</td><td colspan="4">1 个时钟</td><td colspan="4">1 个时钟</td><td colspan="4"></td><td colspan="4">f<sub>XP</sub>/8f<sub>SUB</sub> 个时钟</td></tr>
<tr><td>1</td><td>×</td><td>×</td><td>×</td><td colspan="4">2 个时钟</td><td colspan="4">2 个时钟</td><td colspan="4">2 个时钟</td><td colspan="4">2 个时钟</td><td colspan="4">2 个时钟</td><td colspan="4"></td></tr>
</table>

注意事项 **主系统时钟周期分频因子(PCC0 到 PCC2)的选择，以及从主系统时钟到副系统时钟的切换(CSS 从 0 变到 1)不能同时设置。**

**但对于主系统时钟周期分频因子(PCC0 到 PCC2)的选择，以及副系统时钟到主系统时钟的切换(CSS 从 1 变到 0)可以同时设置。**

备注 **1.** 表 5-7 列出的时钟数是切换前的 CPU 时钟数。

<R> **2.** 将 CPU 时钟从主系统时钟切换到副系统时钟，可通过舍入一个时钟并舍弃小数部分来计算时钟数量(如下所示)。

示例： 将 CPU 时钟从 $f_{SUB}/2$ 切换到 $f_{XP}/2$ (@ 振荡频率 $f_{SUB}$ = 32.768 kHz, $f_{XP}$ = 10 MHz)

$f_{XP}/f_{SUB} = 10000/32.768 \cong 305.1 \rightarrow 306$ 个时钟

通过设置主时钟模式寄存器(MCM)的第 0 位(MCM0)，可以切换主系统时钟(在内部高速振荡时钟和高速系统时钟之间)。

修改 MCM0 后，实际的切换操作不会立即执行；使用切换前的时钟继续操作几个时钟(见 **表 5-8**)。

可以通过 MCM 寄存器的第 1 位(MCS)来确定 CPU 使用内部高速振荡时钟还是高速系统时钟。

表 **5-8.** 主系统时钟切换所需的最长时间

| 切换前设置值 | 切换后设置值 | |
|---|---|---|
| MCM0 | MCM0 | |
| | 0 | 1 |
| 0 | | 1 + $2f_{RH}/f_{XH}$ 时钟 |
| 1 | 1 + $2f_{XH}/f_{RH}$ 时钟 | |

**注意事项　当内部高速振荡时钟切换到高速系统时钟时，MCM 的第 2 位(XSEL)必须预先被设置为 1。XSEL 值在复位释放后只能被修改一次。**

备注　**1.** 表 5-8 所列时钟数是切换之前的主系统时钟数。
**2.** 通过舍去小数部分，计算表 5-8 的时钟数。

示例：　将主系统时钟从内部高速振荡时钟切换到高速系统时钟(@振荡频率 $f_{RH}$ = 8 MHz, $f_{XH}$ = 10 MHz)

$1 + 2f_{RH}/f_{XH} = 1 + 2 \times 8/10 = 1 + 2 \times 0.8 = 1 + 1.6 = 2.6 \rightarrow 2$ 个时钟

### 5.6.9 时钟振荡停止前的状况

下表列出了停止时钟振荡的寄存器标志位设置(禁止外部时钟输入)和时钟振荡停止前的状况。

表 **5-9.** 时钟振荡停止前的状况及标志位设置

| 时钟 | 时钟振荡停止前的状况<br>(禁止外部时钟输入) | SFR 寄存器的标志位设置 |
|---|---|---|
| 内部高速振荡时钟 | MCS = 1 或 CLS = 1<br>(CPU 不使用内部高速振荡时钟) | RSTOP = 1 |
| X1 时钟 | MCS = 0 或 CLS = 1<br>(CPU 不使用高速系统时钟) | MSTOP = 1 |
| 外部主系统时钟 | | |
| XT1 时钟 | CLS = 0<br>(CPU 不使用副系统时钟) | OSCSELS = 0 |
| 外部副系统时钟 | | |

### 5.6.10 外部硬件与源时钟

下表列出了 78K0/KC2 中包含的外部硬件与源时钟。

**表 5-10. 外部硬件与源时钟**

| 源时钟<br>外部硬件 | | 外部硬件时钟 ($f_{PRS}$) | 副系统时钟 ($f_{SUB}$) | 内部低速振荡时钟 ($f_{RL}$) | TM50 输出 | 来自外部硬件引脚的外部时钟 |
|---|---|---|---|---|---|---|
| 16 位定时器/事件计数器 00 | | Y | N | N | N | Y (TI000 引脚)[注1] |
| 8 位定时器/事件计数器 | 50 | Y | N | N | N | Y (TI50 引脚)[注1] |
| | 51 | Y | N | N | N | Y (TI51 引脚)[注1] |
| 8 位定时器 | H0 | Y | N | N | Y | N |
| | H1 | Y | N | Y | N | N |
| 钟表定时器 | | Y | Y | N | N | N |
| 看门狗定时器 | | N | N | Y | N | N |
| 时钟输出[注2] | | Y | Y | N | N | N |
| A/D 转换器 | | Y | N | N | N | N |
| 串行接口 | UART0 | Y | N | N | Y | N |
| | UART6 | Y | N | N | Y | N |
| | CSI10 | Y | N | N | N | Y (SCK10 引脚)[注1] |
| | IIC0 | Y | N | N | N | Y (EXSCL0, SCL0 引脚)[注1] |

注　**1.** 当 CPU 使用副系统时钟且内部高速振荡时钟已经停止时，不要启动依赖由外部硬件引脚输入外部时钟的这些功能。

　　**2.** 仅 48 引脚产品。

备注　Y: 可选, N: 不可选

## 6.1 16 位定时器/事件计数器 00 的功能

16 位定时器/事件计数器 00 具有以下功能。

**(1)** 间隔定时器

16 位定时器/事件计数器 00 和 01 以预置的时间间隔产生中断请求。

**(2)** 方波输出

16 位定时器/事件计数器 00 可以输出任选频率的方波。

**(3)** 外部事件计数器

16 位定时器/事件计数器 00 可以测量外部输入信号的脉冲数。

**(4)** 单脉冲输出

16 位定时器/事件计数器 00 可以输出任意脉冲宽度的单脉冲。

**(5) PPG** 输出

16 位定时器/事件计数器 00 可以输出矩形波，矩形波的频率和输出宽度可以自由设置。

**(6)** 脉冲宽度测量

16 位定时器/事件计数器 00 可以测量外部输入信号的脉冲宽度。

## 6.2 16 位定时器/事件计数器 00 的配置

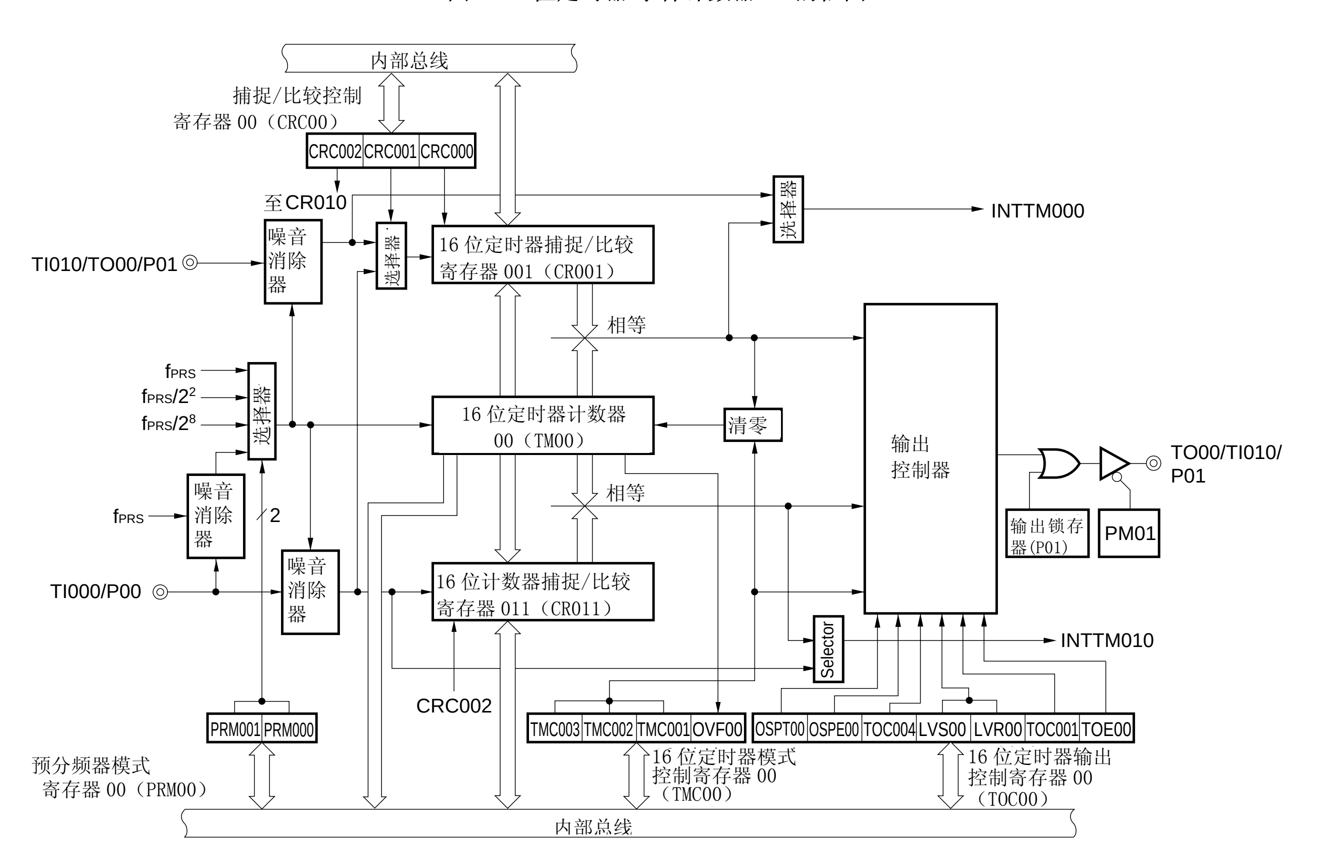
16 位定时器/事件计数器 00 包括以下硬件。

**表 6-1. 16 位定时器/事件计数器 00 的配置**

| 项目 | 配置 |
| --- | --- |
| 定时器/计数器 | 16 位定时器计数器 00 (TM00) |
| 寄存器 | 16 位定时器捕捉/比较寄存器 000, 010 (CR000, CR010) |
| 定时器输入 | TI000, TI010 引脚 |
| 定时器输出 | TO00 引脚, 输出控制器 |
| 控制寄存器 | 16 位定时器模式控制寄存器 00 (TMC00)<br>16 位定时器捕捉/比较控制寄存器 00 (CRC00)<br>16 位定时器输出控制寄存器 00 (TOC00)<br>预分频器模式寄存器 00 (PRM00)<br>端口模式寄存器 0 (PM0)<br>端口寄存器 0 (P0) |

图 6-1 显示了 16 位定时器/事件计数器的框图。

**图 6-1. 位定时器/事件计数器 00 的框图**




注意事项　**1.　TI010** 的有效沿和定时器输出 **(TO00)** 不能够被 **P01** 引脚同时使用。可任选其中一个功能。

**注意事项 2. 如果 16 位定时器模式控制寄存器 00(TMC00)的第 2 和 3 位 2 (TMC002 和 TMC003) 清零为 00，并且输入触发信号，则捕捉数据不确定。**

**3. 为了将捕捉模式改变为比较模式，首先要将 TMC003 和 TMC002 位清零 00，然后改变设置。除非设备复位否则捕捉的值将一直保存在 CR000 中。如果已经改变为比较模式，请确保设定比较值。**

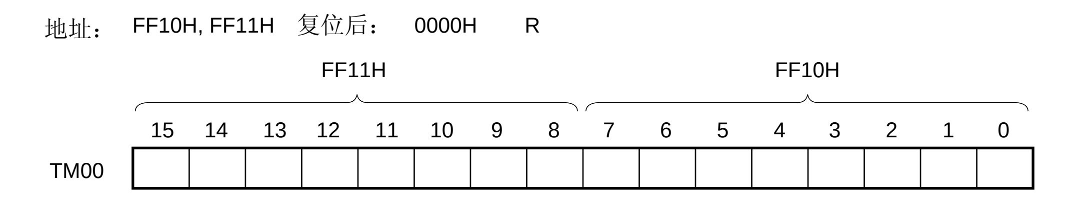
**(1) 16 位定时器计数器 00 (TM00)**

TM00 是 16 位只读寄存器，用于脉冲计数。

计数器随计数时钟的上升沿次数同步增加。

如果在操作期间读取计数值，暂时停止计数时钟输入，则此时的计数值被读取。

**图 6-2. 16 位定时器计数器 00 (TM00) 的格式**




当 16 位定时器模式控制寄存器 00 (TMC00)的第 3 和 2 位不是 00 时，可通过读取 TM00 来获取计数值。如果在 TMC003 和 TMC002 = 00 时读取 TM00，则获取的计数值为 0000H。

如出现以下情况，计数值复位为 0000H。

- 复位信号的产生
- TMC003 和 TMC002 被清零
- 在 TI000 引脚输入有效沿产生清零和启动模式下，TI000 引脚输入有效沿时
- 在 TM00 和 CR000 相等时产生清零和启动模式下，如果 TM00 和 CR000 相等时
- 在单脉冲输出模式下如果 OSPT00 被设置为 1 或 TI000 引脚输入有效沿时

**注意事项 1. 即使读取了 TM00，也不能通过 CR010 捕捉到该值。**

**2．读取 TM00 时，暂停计数时钟输入并在读取定时器后再恢复操作。因此不会出现时钟丢失。**

**(2) 16 位定时器捕捉/比较寄存器 000 (CR000) , 16 位定时器捕捉/比较寄存器 010 (CR010)**

CR000 与 CR010 是 16 位寄存器，具有捕捉或比较功能(通过 CRC00 进行选择)。

当定时器停止操作时(TMC003 与 TMC002 = 00)，修改 CR000 的值。

如果已按指定方式对 CR010 进行设置，则在操作期间可以修改 CR010。如需了解详细信息，可参见 **6.5.1 TM00 操作期间重写 CR010**。

可按 16 位读写这些寄存器。

复位信号的产生可将这些寄存器设置为 0000H。

**图 6-3. 16 位定时器捕捉/比较寄存器 000 (CR000)的格式**

地址： FF12H, FF13H　复位后： 0000H　R/W

| | FF13H | | | | | | | | FF12H | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CR000 | | | | | | | | | | | | | | | | |

**(i) CR000** 用作比较寄存器

CR000 的设置值与 16 位定时器计数器 00 (TM00)的计数值频繁地相比较，如果相等将产生一个中断请求(INTTM000)。在 CR000 被重写前一直保持这个值。

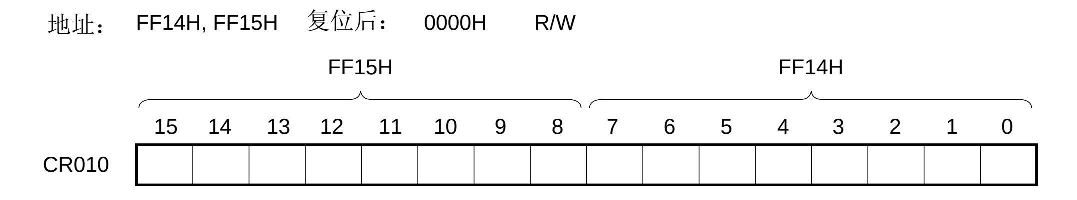
**注意事项　当 CR000 在比较模式时，即使输入触发信号，它也不会执行捕捉操作。**

**(ii) CR000** 用作捕捉寄存器

当输入捕捉触发时，TM00 的计数值被捕捉到 CR000。

可以使用 CRC00 或 PRM00 选择 TI000 引脚的反向脉冲沿或 TI010 引脚的有效沿作为捕捉触发。

**图 6-4. 16 位定时器捕捉/比较寄存器 010 (CR010)的格式**




**(i) CR010** 用作比较寄存器

CR010 的设置值与 TM00 的计数值频繁地相比较，如果相等将产生一个中断请求(INTTM010)。

**注意事项　当 CR010 在比较模式时，即使输入触发信号，它也不会执行捕捉操作。**

**(ii) CR010** 用作捕捉寄存器

当输入捕捉触发时，TM00 的计数值被捕捉到 CR000。

可以选择 TI000 引脚的有效沿作为捕捉触发。TI000 有效沿由 PRM00 来设置。

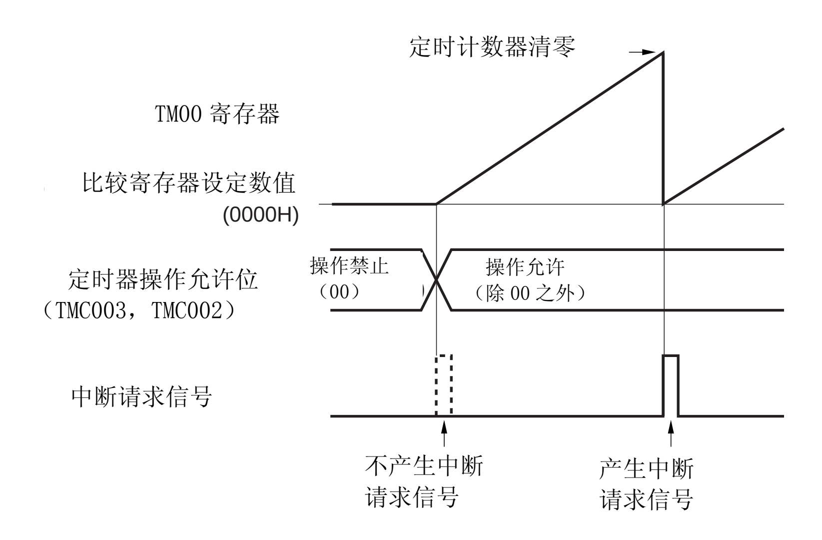
<R> **(iii) 当 CR000 或 CR010 用于比较寄存器时的设定范围**

当 CR000 或 CR010 用于比较寄存器，设定如下。

| 操作 | CR000 寄存器设定范围 | CR010 寄存器设定范围 |
|---|---|---|
| 作为间隔定时器 | 0000H < N ≤ FFFFH | 0000H[注] ≤ M ≤ FFFFH<br>通常不使用此设置。屏蔽相等中断信号 (INTTM010)。 |
| 作为方波输出 | | |
| 作为外部事件计数器 | | |
| 由 TI000 引脚有效沿输入进入清零&启动模式 | 0000H[注] ≤ N ≤ FFFFH | 0000H[注] ≤ M ≤ FFFFH |
| 作为自由运行定时器 | | |
| 作为 PPG 输出 | M < N ≤ FFFFH | 0000H[注] ≤ M < N |
| 作为单脉冲输出 | 0000H[注] ≤ N ≤ FFFFH (N ≠ M) | 0000H[注] ≤ M ≤ FFFFH (M ≠ N) |

注　当设置为 0000H 时，一个相等中断在定时器工作后不会产生，定时器输出也不会变化，第一个相等时序的产生如下所示。相等中断在定时器计数器(TM00 寄存器)从 0000H 变化到 0001H 时产生。

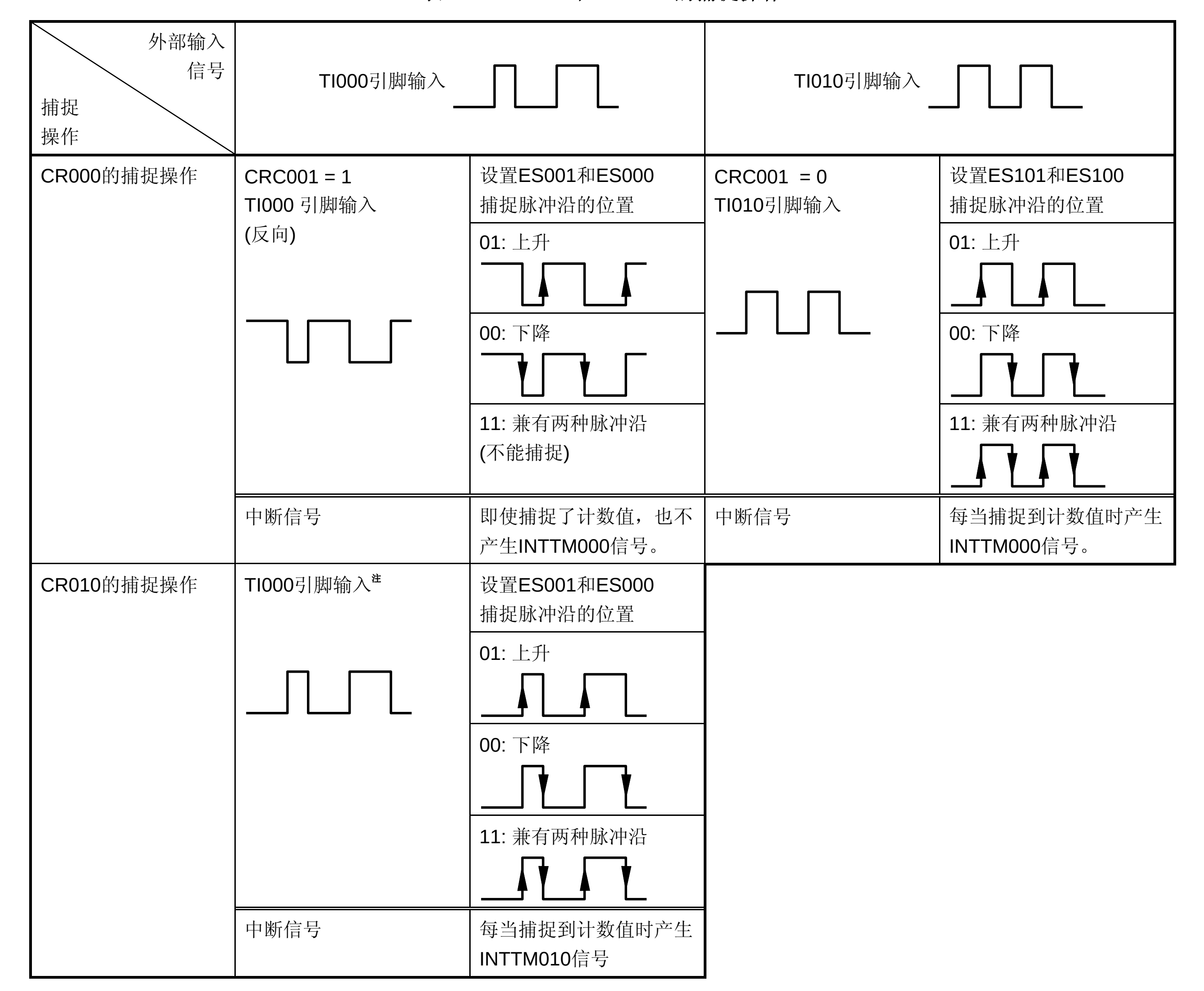
- 当定时器计数器由于溢出被清零时
- 当定时器计数器由于 TI000 引脚有效沿输入清零时(当通过 TI000 引脚有效沿输入进入清零&启动模式时)
- 当由于比较相等定时器计数器被清零时(当由 TM00 和 CR000 相等产生时 (CR000 ≠0000H, CR010 = 0000H)进入清零&启动模式时)




备注　**1.** N: CR000 寄存器设置值，M：CR010 寄存器设置值

**2.** 如需了解 TMC003 和 TMC002 的详细资料，请参考 **6.3 (1) 16 位定时器模式控制寄存器 00(TMC00)**。

**表 6-2. CR000 和 CR010 的捕捉操作**



| 捕捉操作 \ 外部输入信号 | TI000引脚输入 | | TI010引脚输入 | |
|---|---|---|---|---|
| CR000的捕捉操作 | CRC001 = 1<br>TI000 引脚输入<br>(反向) | 设置ES001和ES000<br>捕捉脉冲沿的位置 | CRC001 = 0<br>TI010引脚输入 | 设置ES101和ES100<br>捕捉脉冲沿的位置 |
| | | 01: 上升 | | 01: 上升 |
| | | 00: 下降 | | 00: 下降 |
| | | 11: 兼有两种脉冲沿<br>(不能捕捉) | | 11: 兼有两种脉冲沿 |
| | 中断信号 | 即使捕捉了计数值，也不产生INTTM000信号。 | 中断信号 | 每当捕捉到计数值时产生INTTM000信号。 |
| CR010的捕捉操作 | TI000引脚输入[注] | 设置ES001和ES000<br>捕捉脉冲沿的位置 | | |
| | | 01: 上升 | | |
| | | 00: 下降 | | |
| | | 11: 兼有两种脉冲沿 | | |
| | 中断信号 | 每当捕捉到计数值时产生INTTM010信号 | | |

注　　CRC00 1 位的设置不会对 CR010 的捕捉操作产生影响。

**注意事项** 如果要使用输入到 **TI000** 引脚的反向脉冲沿来捕捉 **TM00** 的计数值，并存入 **CR000**，则在捕捉了计数值后，不产生中断请求信号**(INTTM000)**。如果在操作期间检测到 **TI010** 的有效沿，则不执行捕捉操作，但会产生 **INTTM000** 信号作为外部中断信号。当不使用外部中断时，可以屏蔽 **INTTM000** 信号。

备注　　CRC001: 参见 **6.3 (2) 捕捉/**比较控制寄存器 **00 (CRC00)**
ES101, ES100, ES001, ES000: 参见 **6.3 (4)** 预分频器模式寄存器 **00 (PRM00)**

## 6.3 控制 16 位定时器/事件计数器 00 的寄存器

控制 16 位定时器/事件计数器 00 的寄存器显示如下。

- 16 位定时器模式控制寄存器 00 (TMC00)
- 捕捉/比较控制寄存器 00 (CRC00)
- 16 位定时器输出控制寄存器 00 (TOC00)
- 预分频器模式寄存器 00 (PRM00)
- 端口模式寄存器 0 (PM0)
- 端口寄存器 0 (P0)

### (1) 16 位定时器模式控制寄存器 00 (TMC00)

TMC00 是一个 8 位寄存器，用于设置 16 位定时器操作模式、16 位定时器计数器 00(TM00)清零模式和输出定时，及溢出检测。

操作期间(当 TMC003 与 TMC002 不等于 00)禁止重写 TMC00。当 TMC003 与 TMC002 清零(00)(停止操作)并且 OVF00 也清零(0)时，可以修改它的值。

可以由 1 位或 8 位存储器操作指令设置。

复位信号的产生将 TMC00 清零(00H)。

注意事项　**当设置 TMC002 与 TMC003 为 00（操作停止模式）以外的值时，16 位定时器计数器 00（TM00）开始计数。若要停止操作，则设置 TMC002 和 TMC003 为 00。**

**图 6-5. 16 位定时器模式控制寄存器 00 (TMC00)的格式**

地址: FFBAH 复位后: 00H R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| TMC00 | 0 | 0 | 0 | 0 | TMC003 | TMC002 | TMC001 | OVF00 |

<R>

| TMC003 | TMC002 | 允许16位定时器/事件计数器00的操作 |
|---|---|---|
| 0 | 0 | 禁止TM00的操作。停止供应操作时钟。异步复位内部电路 |
| 0 | 1 | 自由运行定时器模式 |
| 1 | 0 | 通过TI000引脚有效沿输入进入[注] 清零&启动模式 |
| 1 | 1 | TM00和CR000相等时进入清零&启动模式 |

| TMC001 | 反转定时器输出(TO00)的条件 |
|---|---|
| 0 | • TM00与CR000相等或TM00与CR010相等 |
| 1 | • TM00与CR000相等或TM00与CR010相等<br>• TI000引脚有效沿的触发输入 |

| OVF00 | TM00溢出标志 |
|---|---|
| 清零 (0) | 将OVF00清零(0)或 TMC003与TMC002 = 00 |
| 设置 (1) | 发生溢出 |

在所有操作模式下(自由运行定时器模式、清零&启动模式(通过TI000引脚有效沿输入进入)和清零&启动模式(在TM00与CR000相等时进入))当TM00的值由FFFFH变为0000H时，OVF00标志位将被设置为1。
也可以将1写入OVF00进行设置。

注 通过预分频器模式寄存器 00 (PRM00)的第 5 和 4 位(ES001, ES000)设置 TI000 引脚的有效沿。

**(2)　捕捉/比较控制寄存器 00 (CRC0n)**

CRC0n 用于控制 CR00n 与 CR01n 的操作。

操作期间(TMC0n3 与 TMC0n2 不等于 00 时)禁止修改 CRC0n。

可以由 1 位或 8 位存储器操作指令设置 CRC0n。

复位信号的产生将 CRC0n 清零(00H)。

备注　　n = 0, 1

**图 6-6. 捕捉/比较控制寄存器 00 (CRC00)的格式**

地址: FFBCH　复位后: 00H　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CRC00 | 0 | 0 | 0 | 0 | 0 | CRC002 | CRC001 | CRC000 |

| CRC002 | CR010 操作模式的选择 |
|---|---|
| 0 | 用作比较寄存器 |
| 1 | 用作捕捉寄存器 |

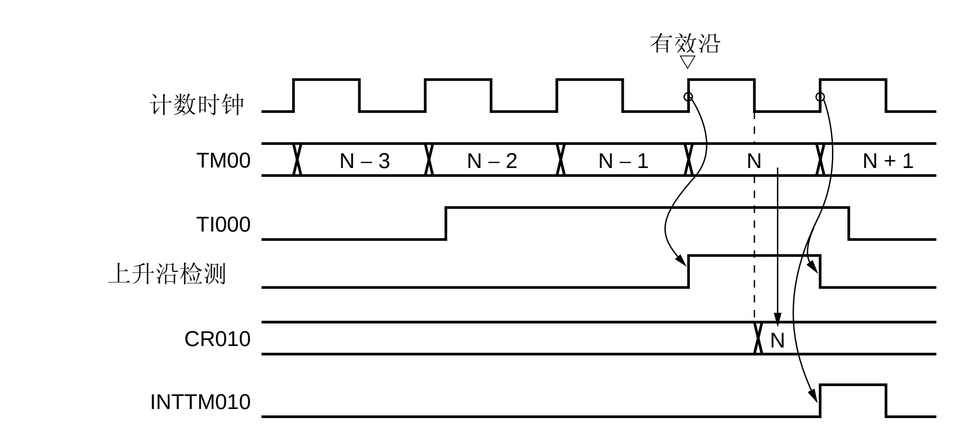
| CRC001 | CR000 捕捉触发选择 |
|---|---|
| 0 | 在 TI010 引脚的有效沿捕捉 |
| 1 | 在TI000 引脚的有效沿反向[注]时捕捉 |
| 由 PRM00 设置 TI010 与 TI000 引脚的有效沿<br>如果在 CRC001=1 时将 ES001 和 ES000 设置为 11(两种脉冲沿)，则不能检测 TI000 引脚的有效沿 | |

| CRC000 | CR000 操作模式选择 |
|---|---|
| 0 | 用作比较寄存器 |
| 1 | 用作捕捉寄存器 |
| 如果 TMC003 与 TMC002 被设置为 11 (清零&启动模式(在 TM00 与 CR000 相等时进入))，则必须将 CRC000 设置为 0 | |

注　　当检测到 TI000 的有效沿时，不执行捕捉操作，但产生 INTTM000 信号作为外部中断信号。

注意事项　　**为确保捕捉操作正确执行，捕捉触发需要的脉冲应大于两个计数时钟的时钟周期，该计数时钟用预分频器模式寄存器 00 (PRM00)选择。**

**图 6-7. CR01n 捕捉操作示例 (指定上升沿有效)**




**(3) 16 位定时器输出控制寄存器 00 (TOC00)**

TOC00 是一个 8 位寄存器，用于控制 TO00 引脚输出。

仅当 OSPT00 操作时才能重写 TOC00(当 TMC003 与 TMC002 不等于 00)。操作期间禁止重写其它位。

但可以在定时器操作期间对 TOC004 进行重写，作为重写 CR010 的一种方法(参见 **6.5.1　TM00 操作期间 CR010 的重写**)。

可以由 1 位或 8 位存储器操作指令设置 TOC00。

复位信号的产生将 TOC00 清零(00H)。

注意事项　**必须按以下步骤设置 TOC00。**

**<1> 设置 TOC004 与 TOC001 为 1。**

**<2> 仅设置 TOE00=1。**

**<3> 设置 LVS00 或 LVR00 为 1。**

**图 6-8. 16 位定时器输出控制寄存器 00 (TOC00) 的格式**

地址: FFBDH　复位后: 00H　R/W

| 符号 | 7 | <6> | <5> | 4 | <3> | <2> | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| TOC00 | 0 | OSPT00 | OSPE00 | TOC004 | LVS00 | LVR00 | TOC001 | TOE00 |

| OSPT00 | 通过软件进行单脉冲输出触发 |
|---|---|
| 0 | – |
| 1 | 单脉冲输出 |
| 该位的读取值总是为“0”。在所有模式下(单脉冲输出模式除外)，不要将该位设置为 1。<br>如果该位为 1，则将 TM00 清零并启动。 | |

| OSPE00 | 单脉冲输出操作控制 |
|---|---|
| 0 | 连续脉冲输出模式 |
| 1 | 单脉冲输出 |
| 在自由运行定时器模式或清零&启动模式(通过 TI000 引脚有效沿输入进入)下，单脉冲输出可以正确进行。<br>在清零&启动模式(由 TM00 与 CR000 相等进入)下不能输出单脉冲。 | |

| TOC004 | CR010 与 TM00 相等时 TO00 引脚输出控制 |
|---|---|
| 0 | 禁止反转操作 |
| 1 | 允许反转操作 |
| 即使 TOC004 = 0，也产生中断信号 (INTTM010)。 | |

| LVS00 | LVR00 | TO00 引脚输出状态设置 |
|---|---|---|
| 0 | 0 | 无变化 |
| 0 | 1 | TO00 引脚输出的初值为低电平 (将 TO00 引脚输出清零(0)) |
| 1 | 0 | TO00 引脚输出的初值为高电平 (将 TO00 引脚输出设置为 1) |
| 1 | 1 | 禁止设置 |

- LVS00 与 LVR00 可以用来设置 TO00 引脚输出电平的初值。如果不必设初值，则将保持 LVS00 与 LVR00 为 00。
- 当 TOE00 = 1 时，必须设置 LVS00 与 LVR00。
  禁止同时将 LVS00、LVR00 和 TOE00 设置为 1。
- LVS00 与 LVR00 是触发位。通过将这两位设置为 1，可以对 TO00 引脚输出电平的初值进行设置。即使将这两位清零(0)，TO00 引脚的输出也不会受到影响。
- LVS00 与 LVR00 的读取值总是为 0。
- 如需了解 LVS00 与 LVR00 的设置，参见 **6.5.2 LVS00 与 LVR00 的设置。**

| TOC001 | CR000 与 TM00 相等时 TO00 引脚输出控制 |
|---|---|
| 0 | 禁止反转操作 |
| 1 | 允许反转操作 |
| 即使 TOC001 = 0，也产生中断信号(INTTM000) | |

| TOE00 | TO00 引脚输出控制 |
|---|---|
| 0 | 禁止输出 (TO00 引脚输出恒为低电平) |
| 1 | 允许输出 |

**(4)** 预分频器模式寄存器 **00 (PRM00)**

PRM0n 用于设置 16 位定时器计数器 00(TM00)的计数时钟以及 TI000 与 TI010 引脚输入的有效沿。
操作期间(当 TMC003 与 TMC002 不等于 00)禁止重写 PRM00。
可由 1 位或 8 位存储器操作指令设置 PRM00。
复位信号的产生将 PRM00 清零(00H)。

注意事项　**1.** 当 **PRM001** 与 **PRM000** 为 **11** 时**(**指定 **TI000** 引脚的有效沿作为计数时钟**)**，不要进行以下设置。
- 清零**&**启动模式**(**通过 **TI000** 引脚有效沿进入**)**
- 设置 **TI000** 引脚作为捕捉触发

**2.** 如果在 **TI000** 或 **TI010** 引脚为高电平并且指定 **TI000** 或 **TI010** 引脚的有效沿为上升沿或兼有两种脉冲沿时允许 **16** 位定时器**/**事件计数器 **00** 操作，**TI000** 或 **TI010** 引脚的高电平将被检测为上升沿。当 **TI000** 或 **TI010** 引脚被上拉时要注意。然而一旦定时器操作已经被停止然后再次允许时则不检测上升沿。

**3.** 对于 **P01** 引脚，**TI010** 的有效沿和定时器输出**(TO00)**不能同时使用，对于 **P06** 引脚，**TI011** 的有效沿和定时器输出**(TO01)**也不能同时使用。可以任选其一。

图 **6-9.** 预分频器模式寄存器 **00 (PRM00)** 的格式

地址: FFBBH　复位后: 00H　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PRM00 | ES101 | ES100 | ES001 | ES000 | 0 | 0 | PRM001 | PRM000 |

| ES101 | ES100 | TI010 引脚有效沿选择 |
|---|---|---|
| 0 | 0 | 下降沿 |
| 0 | 1 | 上升沿 |
| 1 | 0 | 禁止设置 |
| 1 | 1 | 兼有上升沿和下降沿 |

| ES001 | ES000 | TI000 引脚有效沿选择 |
|---|---|---|
| 0 | 0 | 下降沿 |
| 0 | 1 | 上升沿 |
| 1 | 0 | 禁止设置 |
| 1 | 1 | 兼有上升沿和下降沿 |

| PRM001 | PRM000 | 计数时钟选择 | | | | |
|---|---|---|---|---|---|---|
| | | | $f_{PRS}$ = 2 MHz | $f_{PRS}$ = 5 MHz | $f_{PRS}$ = 10 MHz | $f_{PRS}$ = 20 MHz |
| 0 | 0 | $f_{PRS}$ | 2 MHz | 5 MHz | 10 MHz | 20 MHz |
| 0 | 1 | $f_{PRS}/2^2$ | 500 kHz | 1.25 MHz | 2.5 MHz | 5 MHz |
| 1 | 0 | $f_{PRS}/2^8$ | 7.81 kHz | 19.53 kHz | 39.06 kHz | 78.12 kHz |
| 1 | 1 | TI000 有效沿[注] | | | | |

注　外部时钟脉冲宽度要求大于内部时钟($f_{PRS}$)的两个周期宽度。

备注　$f_{PRS}$: 外围硬件时钟频率

**(5)　端口模式寄存器 0 (PM0)**

该寄存器按位设置端口 0 为输入/输出模式。

如果使用 P01/TO00/TI010 和 P06/TO01/TI011 引脚用于定时器输出，则需要将 PM01 和 PM06 以及 P01 与 P06 的输出锁存器清零。

如果使用 P00/TI000、P01/TO00/TI010、P05/TI001/$\overline{\text{SSI11}}$ 和 P06/TO01/TI011 引脚用于定时器输入，则需要将 PM00、PM01、PM05 和 PM06 设置为 1，此时 P00、P01、P05 和 P06 的输出锁存器可以为 0 也可以为 1。

可由 1 位或 8 位存储器操作指令设置 PM0。

复位信号的产生将 PM0 设置为 FFH。

**图 6-10.　端口模式寄存器 0 (PM0)的格式**

地址：FF20H　　复位后：FFH　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM0 | 1 | PM06 | PM05 | PM04 | PM03 | PM02 | PM01 | PM00 |

| PM0n | P0n 引脚 I/O 模式选择(n=0～6) |
|---|---|
| 0 | 输出模式(输出缓冲器打开) |
| 1 | 输入模式(输出缓冲器关闭) |

## 6.4 16 位定时器/事件计数器 00 和 01 的操作

### 6.4.1 间隔定时器的操作

如果 16 位定时器模式控制寄存器 (TMC00)的第 3 和 2 位(TMC003 和 TMC002)为 11 (清零&启动模式(在 TM00 与 CR000 相等时进入))，则启动计数操作与计数时钟同步。

当之后的 TM00 值与 CR000 值相等时，将 TM00 清零(0000H)并产生一个相等中断信号(INTTM000)。INTTM000 信号允许 TM00 作为间隔定时器操作。

备注 **1.** 如需了解 I/O 引脚的设置，可参见 **6.3 (5) 端口模式寄存器 0 (PM0)**。
**2.** 要了解如何允许 INTTM00n 中断，参见 **第十八章 中断功能**。

**图 6-11. 间隔定时器操作框图**

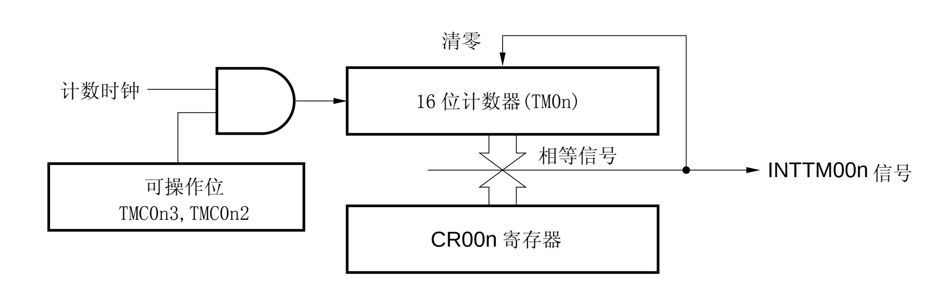



**图 6-12. 间隔定时器操作的基本时序示例**

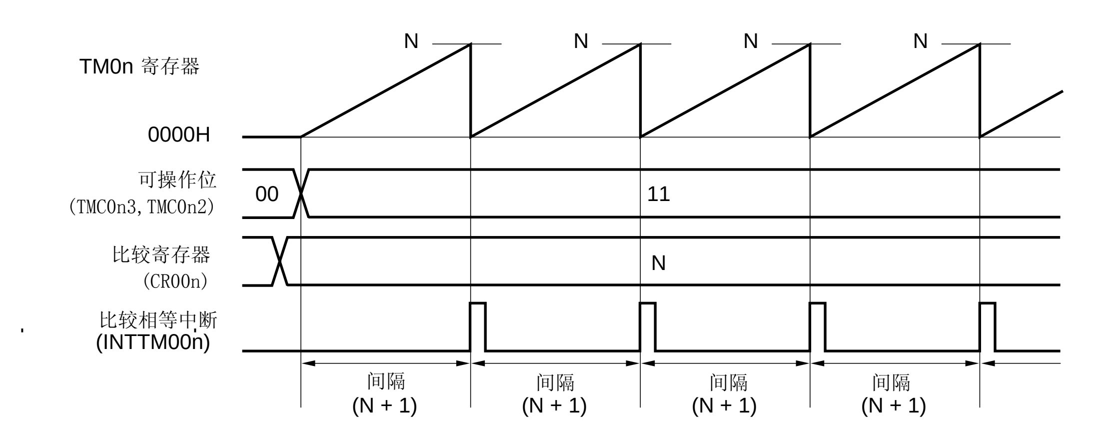


图 6-13. 间隔定时器操作的寄存器设置示例

**(a) 16 位定时器模式控制寄存器 00 (TMC00)**

| | | | | TMC0n3 | TMC0n2 | TMC0n1 | OVF0n |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |

TMC0n3, TMC0n2: TM0n 与 CR00n 相等时清零并启动

**(b) 捕捉/比较控制寄存器 00 (CRC00)**

| | | | | | CRC0n2 | CRC0n1 | CRC0n0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CRC0n0: CR00n 用作比较寄存器

**(c) 16 位定时器输出控制寄存器 00 (TOC00)**

| | OSPT0n | OSPE0n | TOC0n4 | LVS0n | LVR0n | TOC0n1 | TOE0n |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**(d) 预分频器模式寄存器 00 (PRM00)**

| ES1n1 | ES1n0 | ES0n1 | ES0n0 | 3 | 2 | PRM0n1 | PRM0n0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0/1 | 0/1 |

PRM0n1, PRM0n0: 选择计数时钟

**(e) 16 位定时器计数器 00 (TM00)**

通过读取 TM0n，获得计数值。

**(f) 16 位捕捉/比较寄存器 000 (CR000)**

如果 CR000=M，则间隔时间表示如下。

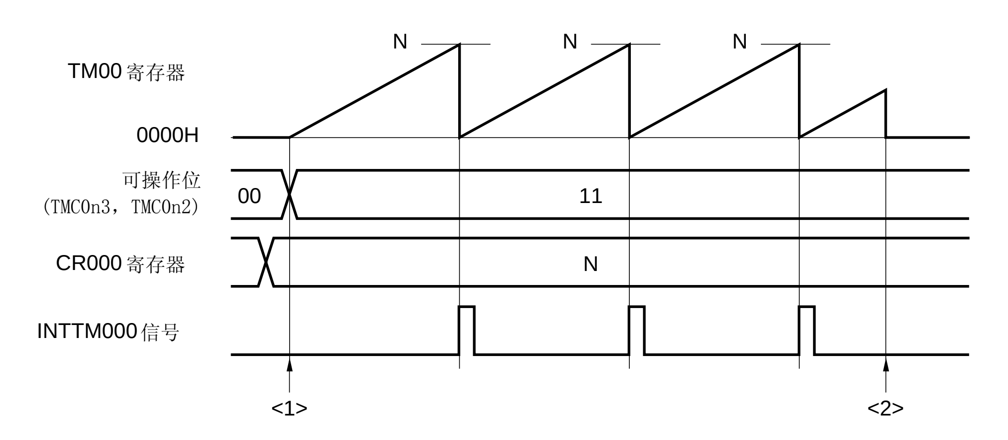
- 间隔时间 = (M + 1) × 计数时钟周期

禁止将 CR000 设置为 0000H。

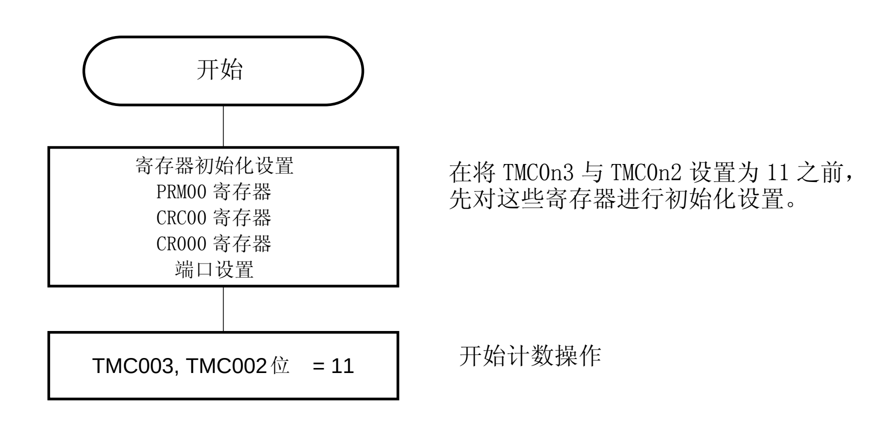
**(g) 16 位捕捉/比较寄存器 010 (CR010)**

通常 CR010 不用作间隔定时器。但当 CR010 的值与 TM00 的值相等时，会产生比较相等中断(INTTM010)。因此通过使用中断屏蔽标志(TMMK010)屏蔽中断请求。

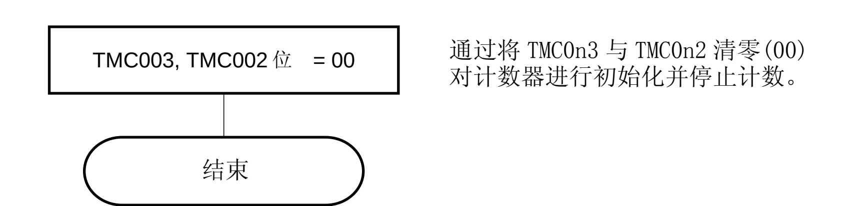
图 6-14. 间隔定时器软件处理示例




<1> 计数操作启动流程




<2> 计数操作停止流程



### 6.4.2 方波输出操作

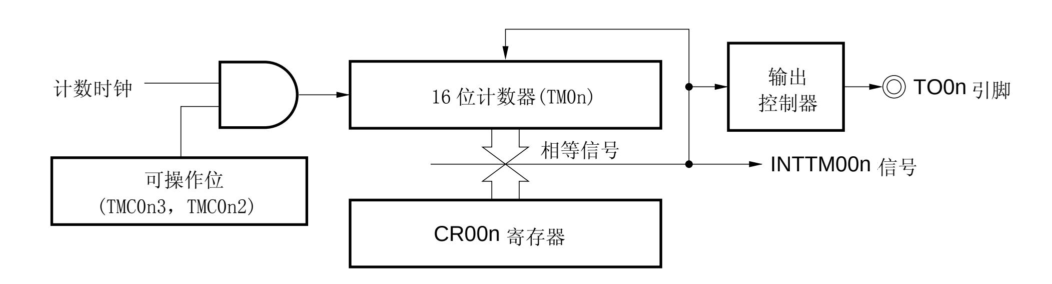
当 16 位定时器/事件计数器 00 用作间隔定时器时(参见 **6.4.1**)，通过将 16 位定时器输出控制寄存器 00 (TOC00)设置为 03H，可以从 TO00 引脚输出一个方波。

当 TMC003 与 TMC002 被设置为 11 时(计数清零&启动模式(在 TM00 与 CR000 相等时进入))，启动计数操作与计数时钟同步。

当之后 TM00 的值与 CR000 的值相等时，将 TM00 清零(0000H)、产生中断信号(INTTM000)并反转 TO00 引脚的输出。反转的 TO00 引脚输出以恒定的间隔允许 TO00 输出一个方波。

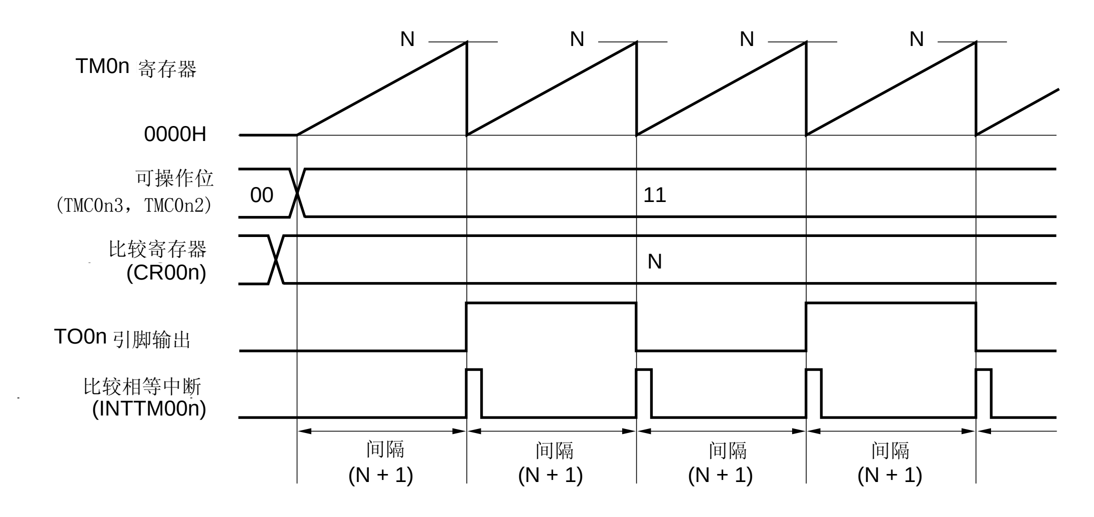
**备注**　**1.** 如需了解 I/O 引脚的设置，可参见 **6.3 (5) 端口模式寄存器 0 (PM0)**。
　　　　**2.** 要了解如何允许 INTTM000 信号中断，参见 **第十八章 中断功能**。

**图 6-15. 方波输出操作框图**




**备注**　n = 0, 1

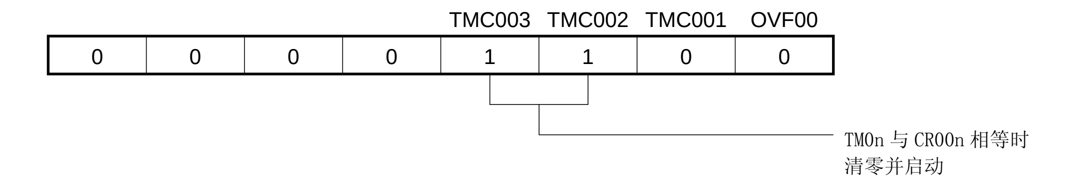
**图 6-16. 方波输出操作的基本时序示例**



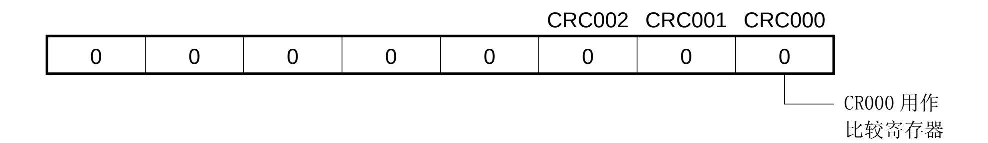

**备注**　n = 0, 1

图 **6-17.** 方波输出操作的寄存器设置示例

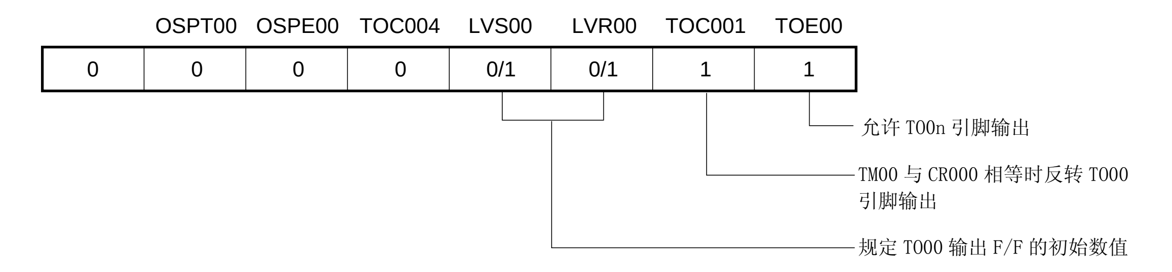
**(a) 16** 位定时器模式控制寄存器 **00 (TMC00)**



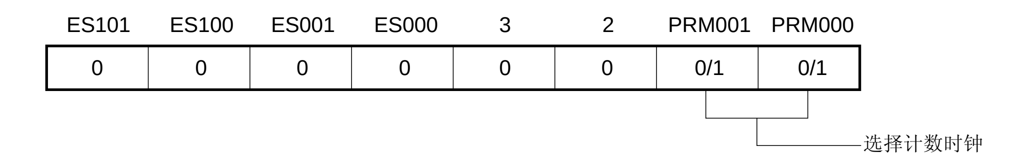

**(b)** 捕捉**/**比较控制寄存器 **00 (CRC00)**




<R> **(c) 16** 位定时器输出控制寄存器 **00 (TOC00)**




**(d)** 预分频器模式寄存器 **00 (PRM00)**




**(e) 16** 位定时器计数器 **00 (TM00)**

通过读取 TM00，获得计数值。

**(f) 16** 位捕捉**/**比较寄存器 **000 (CR000)**

如果 CR000=M，则间隔时间表示如下。

• 方波频率 = 1 / [2 × (M + 1) × 计数时钟周期]

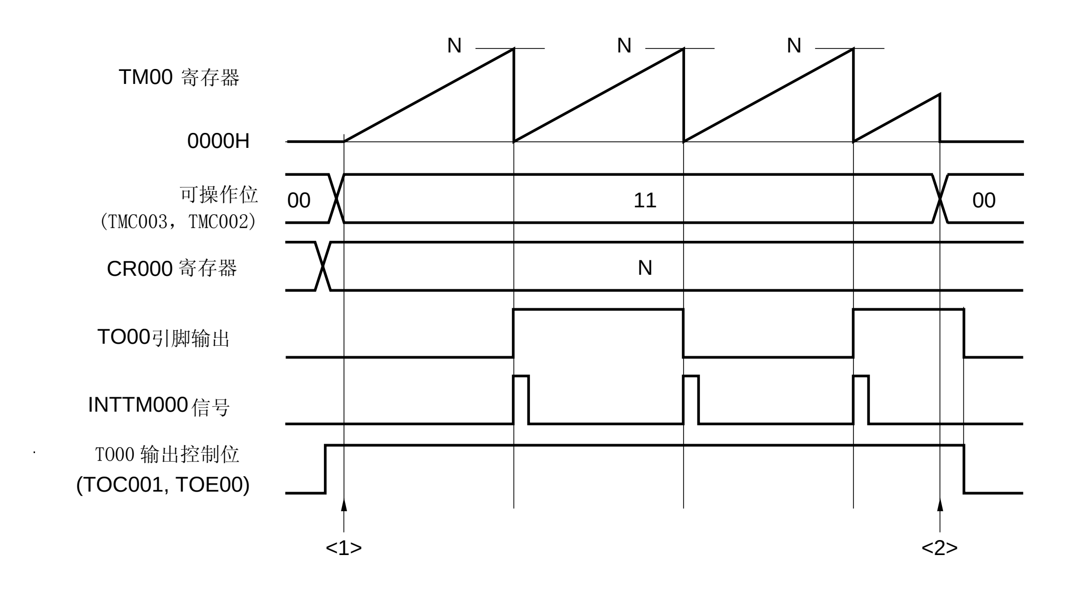
禁止将 CR000 设置为 0000H。

**(g) 16** 位捕捉**/**比较寄存器 **010 (CR010)**

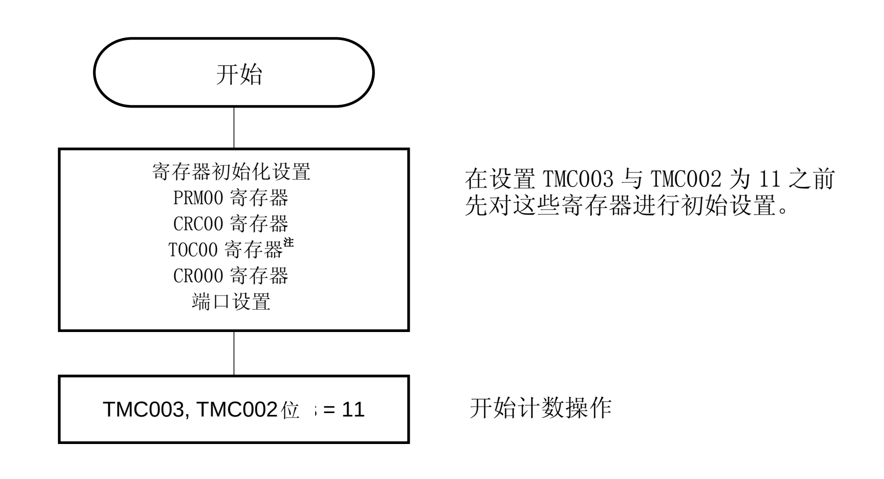
通常 CR010 不用作方波输出。但当 CR010 的值与 TM00 的值相等时，会产生比较相等中断(INTTM010)。因此，通过使用中断屏蔽标志(TMMK010)屏蔽中断请求。

<R>

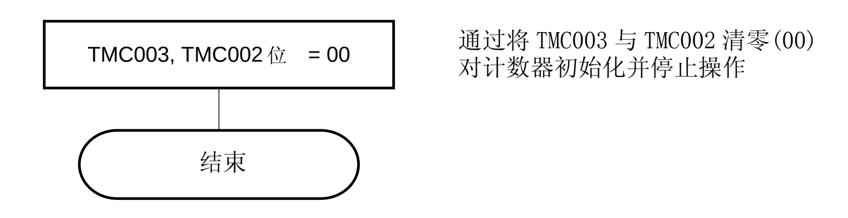
**图 6-18.** 方波输出软件处理示例




<1> 计数操作启动流程




<2> 计数操作停止流程




注　设置 TOC00 时必须特别小心。如需了解详细信息，参见 **6.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)**。

### 6.4.3 外部事件计数器的操作

当预分频器模式寄存器 00 (PRM00)的第 1 和 0 位(PRM001 和 PRM000)被设置为 11(使用 TI000 引脚的有效沿计数)以及 16 位定时器模式控制寄存器 00 (TMC00)的第 3 和 2 位(TMC003 和 TMC002)被设置为 11 时，开始对外部事件输入的有效沿进行计数，并产生用于指示 TM00 和 CR000 之间相等的相等中断信号(INTTM000)。

如果要输入外部事件，可以使用 TI000 引脚。因此定时器/事件计数器不能用作清零&启动模式(通过 TI000 引脚有效沿(TMC003 与 TMC002 = 10)进入)下的外部事件计数器。

INTTM000 信号按下列时序产生。

- INTTM000 信号产生时序 (第 2 次或之后产生)
  = 外部事件输入的有效沿检测次数 × (CR000 的设置值 + 1)

但在定时器/事件计数器开始操作后，将按以下时序立即产生第 1 次相等中断。

- INTTM000 信号产生时序 (仅在第 1 次产生)
  =外部事件输入的有效沿检测次数× (CR000 的设置值 + 2)

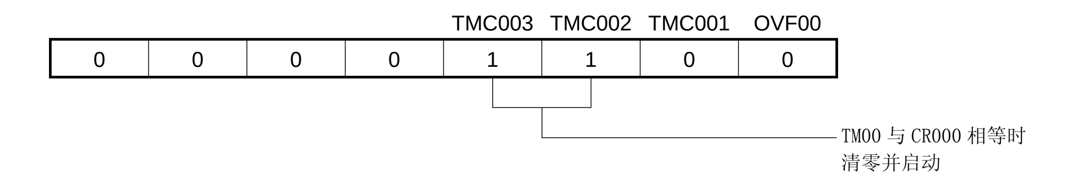
如果要检测有效沿，则应在 $f_{PRS}$ 时钟周期内对输入到 TI000 引脚的信号进行采样。在一个周期内检测到两次脉冲沿时才认为检测到有效沿，这样可以消除一个窄脉冲宽度的噪音。

备注　**1.** 如需了解 I/O 引脚的设置，参见 **6.3 (5) 端口模式寄存器 0 (PM0)**.
　　　**2.** 要了解如何允许 INTTM000 信号中断，参见 **第十八章 中断功能**。

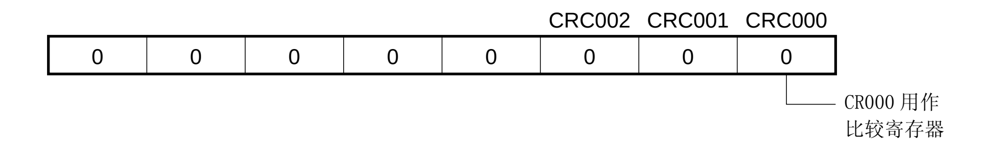
**图 6-19. 外部事件计数器操作框图**

**图 6-20. 外部事件计数器模式下寄存器设置示例 (1/2)**

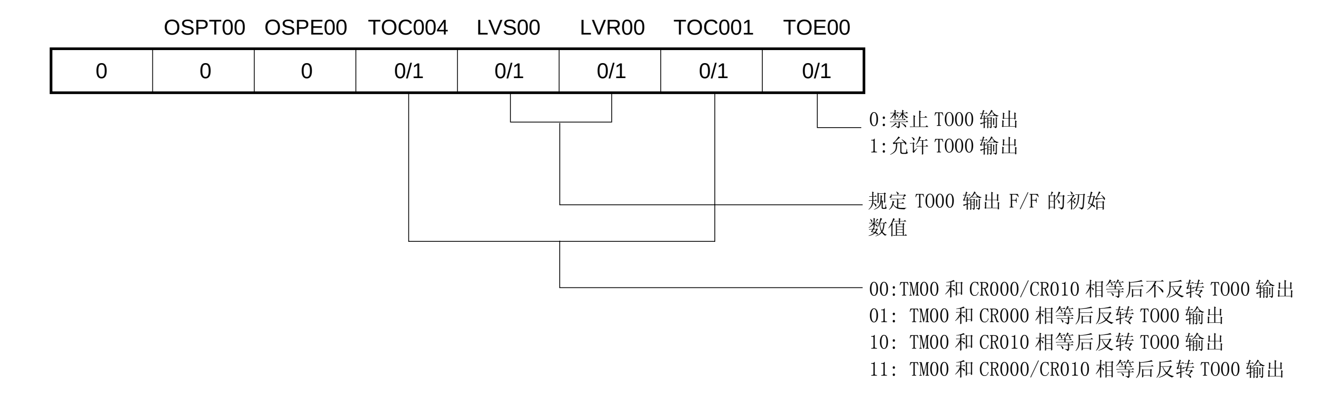
**(a) 16 位定时器模式控制寄存器 00 (TMC00)**



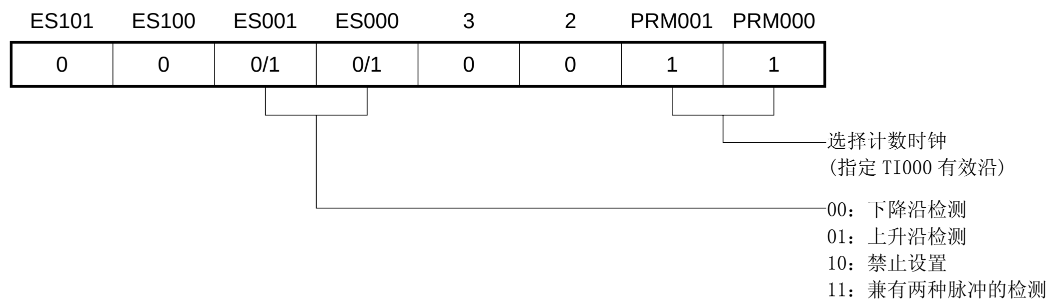

**(b) 捕捉/比较控制寄存器 00 (CRC00)**




<R> **(c) 16 位定时器输出控制寄存器 00 (TOC00)**




**(d) 预分频器模式寄存器 00 (PRM00)**




**(e) 16 位定时器计数器 00 (TM00)**

通过读取 TM00，获得计数值。

**(f) 16 位捕捉/比较寄存器 000 (CR000)**

如果 M=CR000，则当外部事件数=(M + 1)时产生中断信号(INTTM000)。
禁止将 CR000 设置为 0000H。

**图 6-20. 外部事件计数器模式下寄存器设置示例 (2/2)**

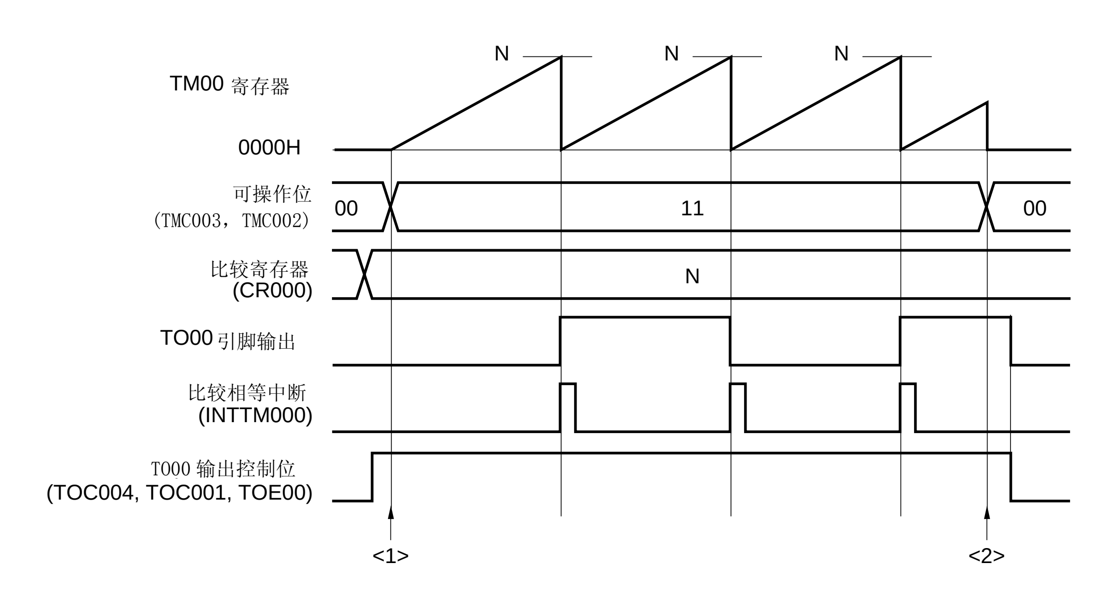
**(g) 16 位捕捉/比较寄存器 010 (CR010)**

通常 CR010 不用于外部事件计数器模式。但当 CR010 的值与 TM00 的值相等时会产生比较相等中断(INTTM010)。

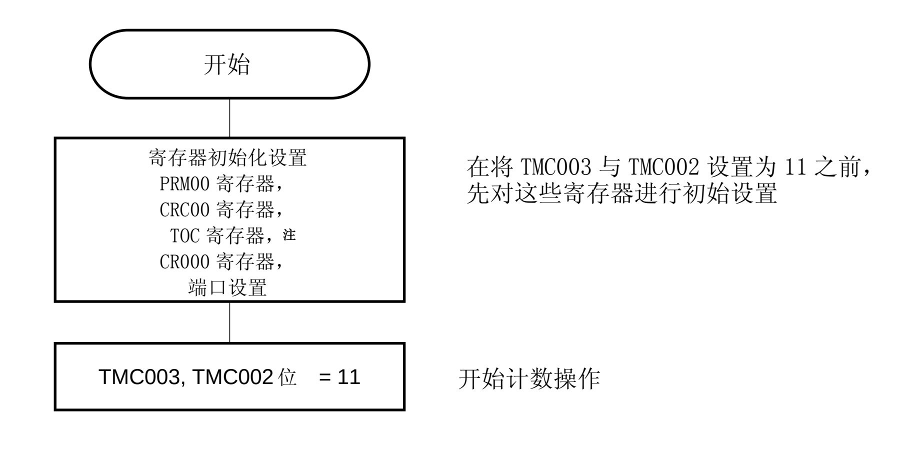
因此，通过使用中断屏蔽标志(TMMK010)屏蔽中断请求。

<R>

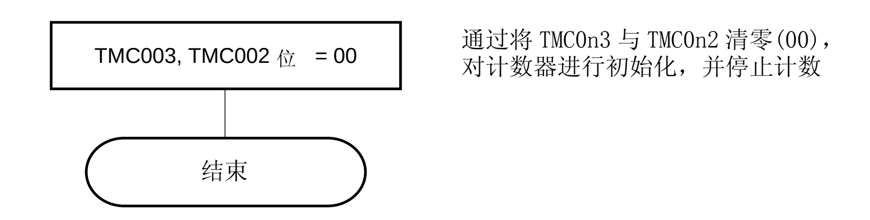
**图 6-21. 外部事件计数器模式下的软件处理示例**




<1>计数操作启动流程




<2>计数操作停止流程




注　设置 TOC00 时必须特别小心。如需了解详细信息，参见 **6.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)**。

### 6.4.4 清零&启动模式(通过 TI000 引脚有效沿输入进入)下的操作

当 16 位定时器模式控制寄存器 00 (TMC00)的第 3 和 2 位(TMC003 和 TMC002)被设置为 10(清零&启动模式(通过 TI000 引脚有效沿输入进入))并且向定时器/事件计数器提供计数时钟(由 PRM00 设置)时，TM00 开始计数。在计数操作期间，如果检测到 TI000 引脚的有效沿，则将 TM00 清零(0000H)并再次开始计数。如果没有检测到 TI000 引脚的有效沿，则 TM00 溢出并继续计数。

可通过 TI000 引脚的有效沿对 TM00 清零。在操作开始后，不会立即启动计数器。

CR00n 与 CR010 可用作比较寄存器和捕捉寄存器。

**(a) CR000 与 CR010 用作比较寄存器**

当 TM00 的值与 CR000、CR010 的值相等时，产生信号 INTTM000 与 INTTM010。

**(b) CR000 与 CR010 用作捕捉寄存器**

当有效沿输入到 TI010 引脚时(或者当反向的有效脉冲沿输入到 TI000 引脚时)，TM00 的计数值被捕捉到 CR000，同时产生 INTTM000 信号。

当有效沿输入到 TI000 引脚时，TM00 的计数值被捕捉到 CR010，同时产生 INTTM010 信号。一旦捕捉了计数值，计数器即被清零(0000H)。

注意事项　**不要将 TI000 引脚的有效沿(PRM001 与 PRM000 = 11)设置为计数时钟。当 PRM001 与 PRM000 = 11 时，将 TM00 清零。**

备注　**1.** 如需了解 I/O 引脚的设置，参见 **6.3 (5) 端口模式寄存器 0 (PM0)**。
　　　**2.** 要了解如何允许 INTTM000 信号中断，参见 **第十八章　中断功能**。

**(1) 清零&启动模式(通过 TI000 引脚有效沿输入进入)下的操作**
**(CR000: 比较寄存器, CR010: 比较寄存器)**

图 **6-22.**　**清零&启动模式(通过 TI000 引脚有效沿输入进入)的框图**
**(CR000: 比较寄存器, CR010: 比较寄存器)**

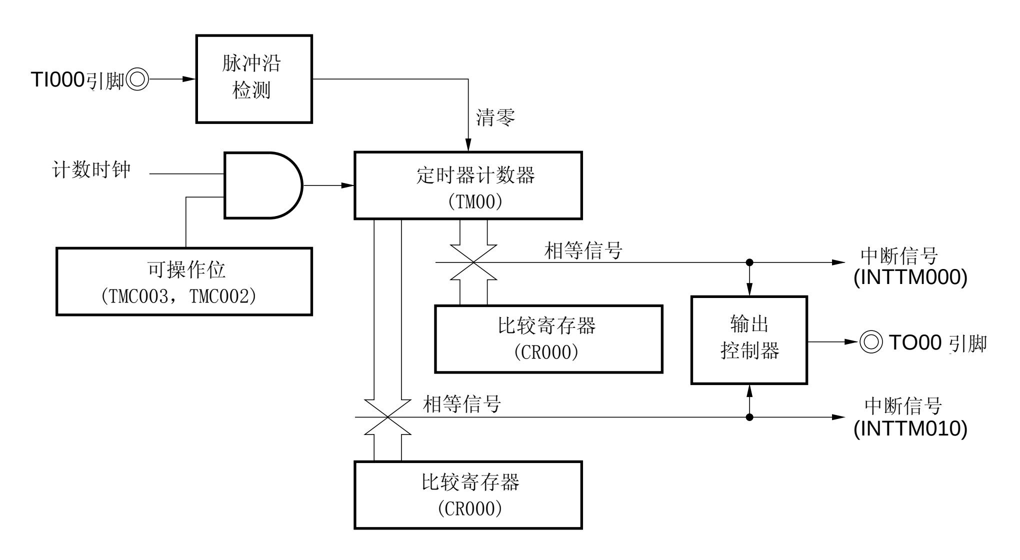
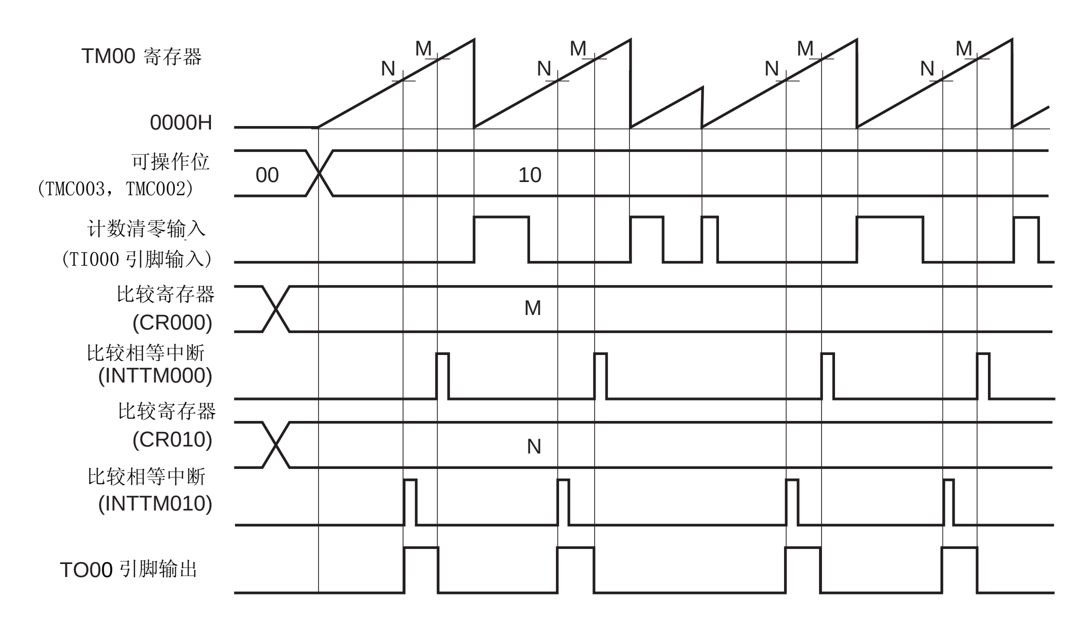
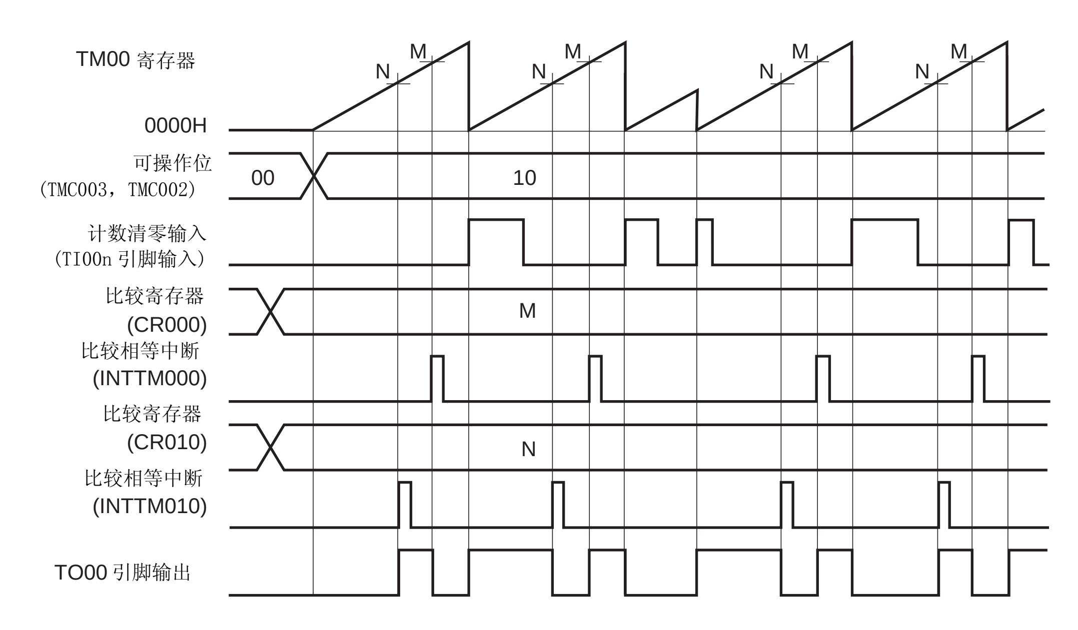


**图 6-23.　清零&启动模式(通过 TI000 引脚有效沿输入进入)时序示例**
**(CR000: 比较寄存器, CR010: 比较寄存器)**

**(a) TOC00 = 13H, PRM00 = 10H, CRC00, = 00H, TMC00 = 08H**




**(b) TOC00 = 13H, PRM00 = 10H, CRC00, = 00H, TMC00 = 0AH**




(a)　与(b) 根据 16 位定时器模式控制寄存器 00 (TMC00) 第 1 位(TMC001) 的设置不同有如下变化。

(a)　当 TM00 与比较寄存器相等时反转 TO00 引脚的输出电平。

(b)　当 TM00 与比较寄存器相等或者检测到 TI000 有效沿时反转 TO00 引脚的输出电平。

**(2) 清零&启动模式(通过 TI000 引脚有效沿输入进入)下的操作**
**(CR000: 比较寄存器, CR010: 捕捉寄存器)**

**图 6-24. 清零&启动模式(通过 TI000 引脚有效沿输入进入)的框图**
**(CR000: 比较寄存器, CR010: 捕捉寄存器)**

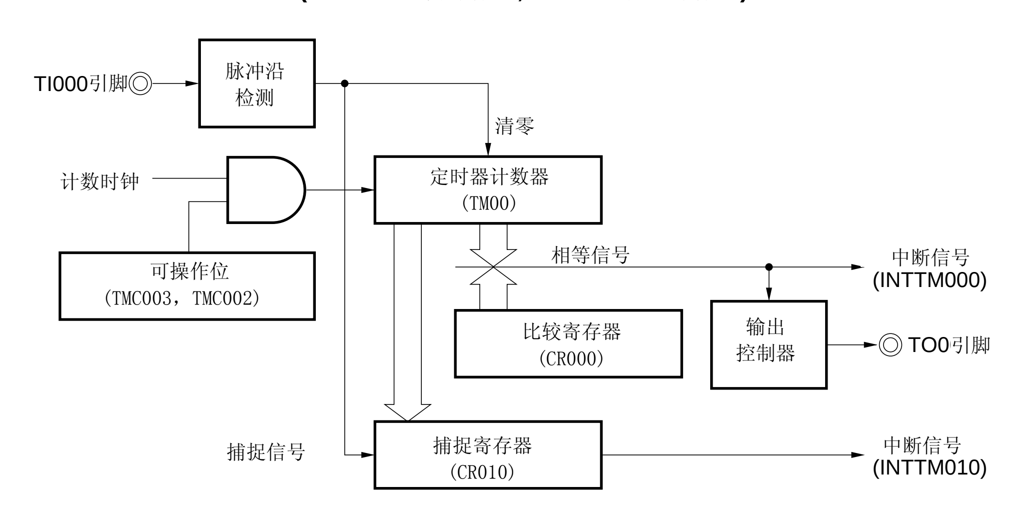
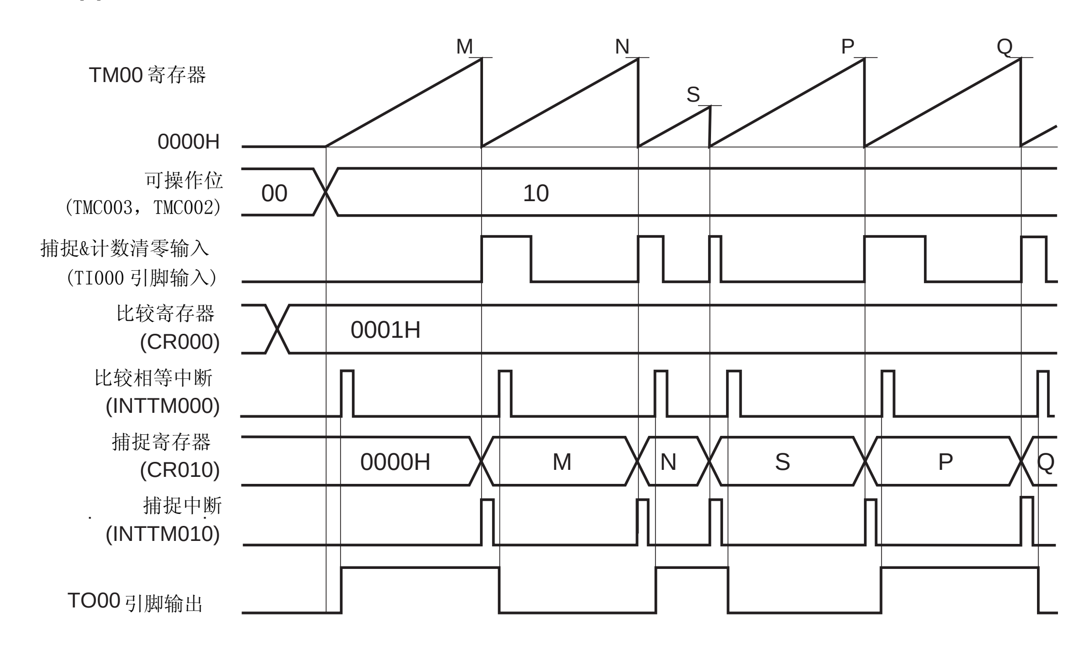



**图 6-25. 清零&启动模式(通过 TI000 引脚有效沿输入进入) 时序示例**
**(CR000: 比较寄存器, CR010: 捕捉寄存器) (1/2)**

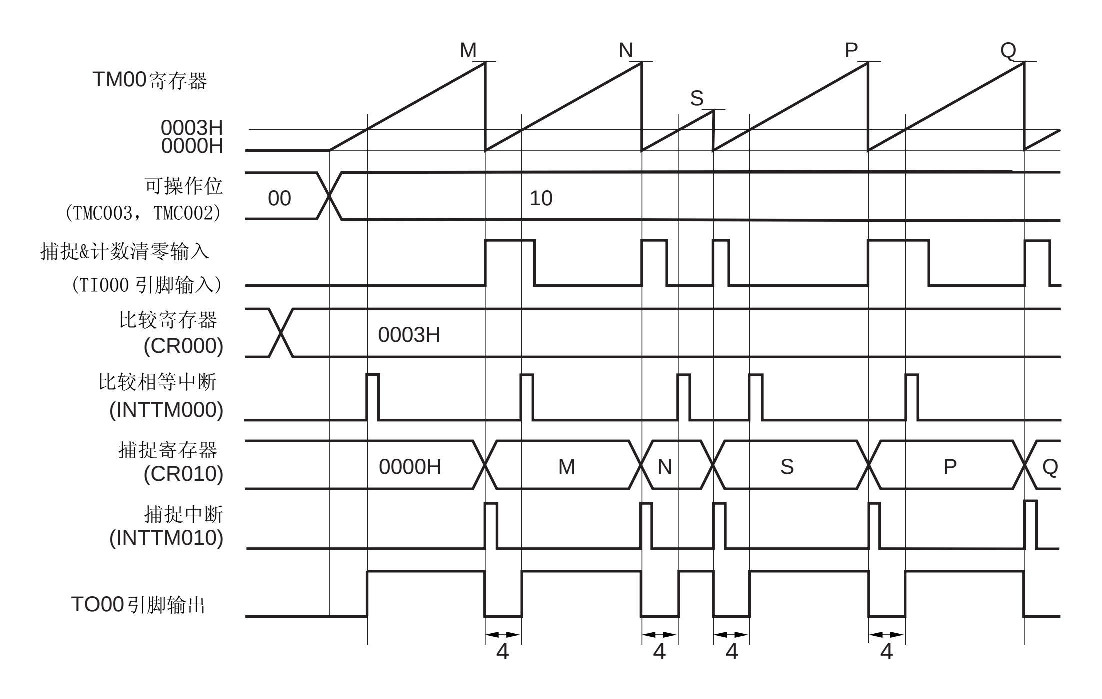
**(a) TOC00 = 13H, PRM00 = 10H, CRC00, = 04H, TMC00 = 08H, CR000 = 0001H**




这是一个应用示例：当捕捉到计数值并清零时反转 TO00 引脚的输出电平。
当检测到 TI000 引脚的有效沿时捕捉计数值存入 CR010 并将 TM00 清零(0000H)。当 TM00 的计数值=0001H 时，产生比较相等中断信号(INTTM000)，并反转 TO00 引脚的输出电平。

**图 6-25.　清零&启动模式(通过 TI000 引脚有效沿输入进入)时序示例**
**(CR000: 比较寄存器, CR010: 捕捉寄存器) (2/2)**

**(b) TOC00 = 13H, PRM00 = 10H, CRC00, = 04H, TMC00 = 0AH, CR000 = 0003H**




这是一个应用示例：当捕捉计数值并清零时从 TO00 引脚输出 CR000 设置的宽度(本例是 4 个时钟)。
当检测到 TI000 引脚的有效沿时，捕捉计数值存入 CR010、产生捕捉中断信号(INTTM010)、将 TM00 清零(0000H)并反转 TO00 引脚的输出电平。当 TM00 的计数值=0003H 时(已经计数 4 个时钟)，产生比较相等中断信号(INTTM000)并反转 TO00 引脚的输出电平。

**(3)　清零&启动模式(通过 TI000 引脚有效沿输入进入)的操作**
**(CR000: 捕捉寄存器, CR010: 比较寄存器)**

图 **6-26.** 清零&启动模式**(**通过 **TI000** 引脚有效沿输入进入**)**的框图
**(CR000:** 捕捉寄存器**, CR010:** 比较寄存器**)**

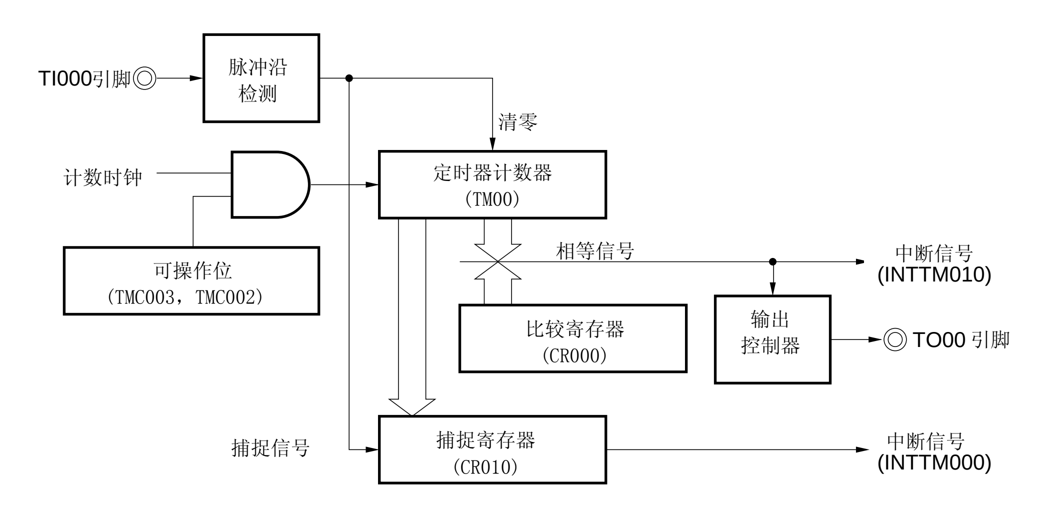
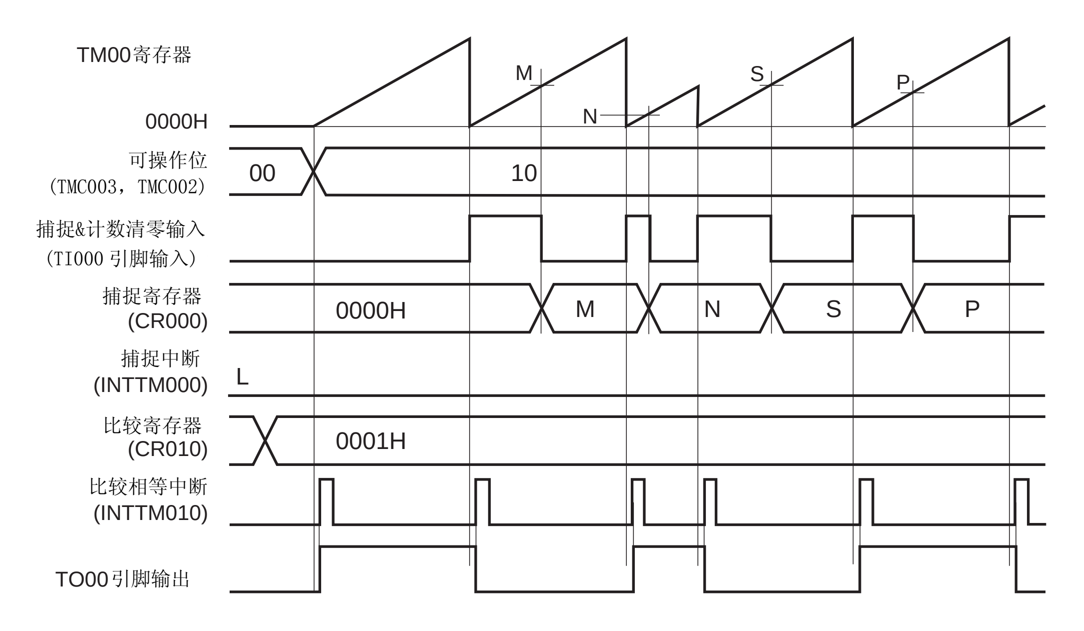


**图 6-27. 清零&启动模式(通过 TI000 引脚有效沿输入进入) 时序示例**
**(CR000: 捕捉寄存器, CR010: 比较寄存器) (1/2)**



**(a) TOC00 = 13H, PRM00 = 10H, CRC00, = 03H, TMC00 = 08H, CR010 = 0001H**




这是一个应用示例：当捕捉计数值并清零时反转 TO00 引脚的输出电平。

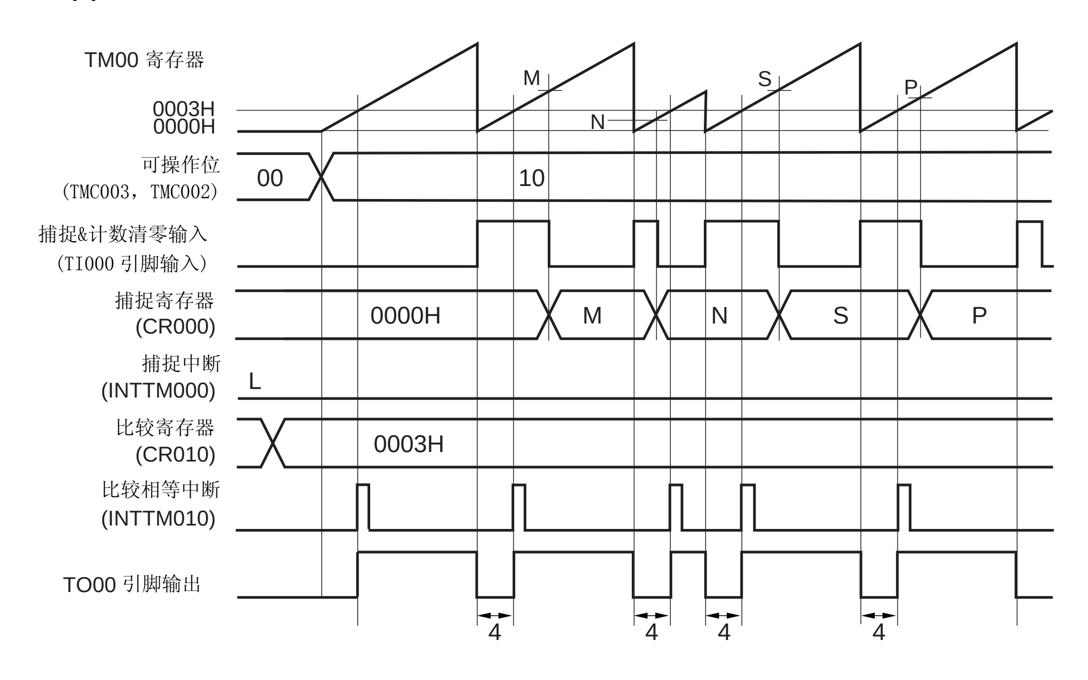
在检测到 TI000 引脚的上升沿时将 TM00 清零，而在检测到 TI000 引脚的下降沿时，TM00 的计数值被捕捉到 CR000。

当捕捉/比较控制寄存器 00 (CRC00)的第 1 位(CRC001)=1 时，TM00 的计数值被捕捉到 CR000(在 TI000 引脚输入信号的反向沿捕捉)，但不产生捕捉中断信号(INTTM000)。然而当检测到 TI010 引脚的有效沿时产生 INTTM000 信号。如果不使用 INTTM000 信号，则将其屏蔽。

**图 6-27. 清零&启动模式(通过 TI000 引脚有效沿输入进入) 时序示例**
**(CR000: 捕捉寄存器, CR010: 比较寄存器) (2/2)**



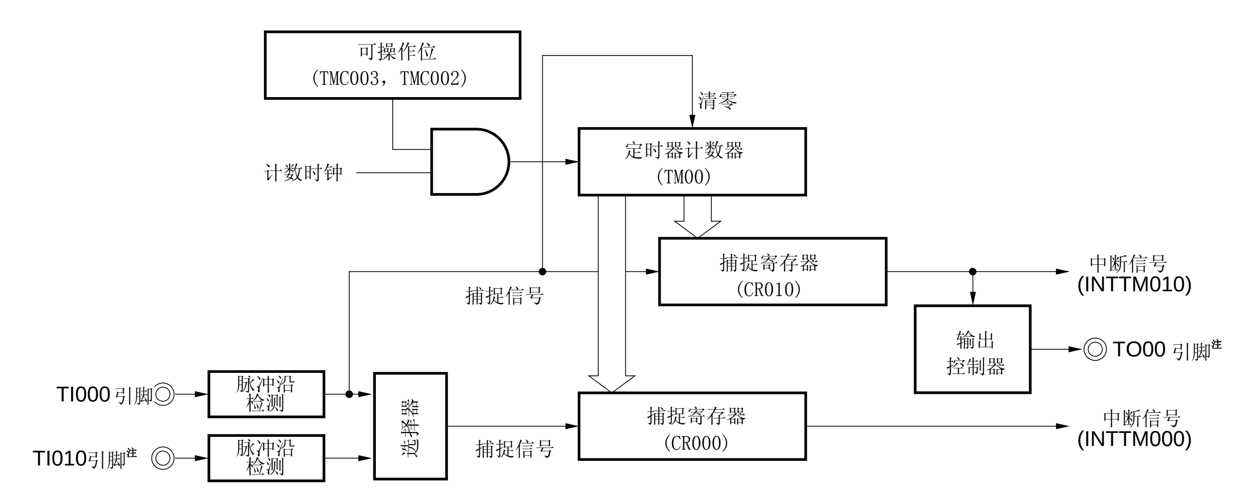
**(b) TOC00 = 13H, PRM00 = 10H, CRC00, = 03H, TMC00 = 0AH, CR010 = 0003H**




这是一个应用示例：当捕捉计数值并清零时从 TO00 引脚输出 CR010 设置的宽度(本例是 4 个时钟)。
在检测到 TI000 引脚的上升沿时将 TM00 清零(0000H)，而在检测到 TI000 引脚的下降沿时，TM00 的计数值被捕捉到 CR000。当 TM00 被清零（0000H）后，TO00 引脚的输出电平被反转，因为 TI00 引脚检测到上升延或者 TM00 的值与比较寄存器（CR010）的值相相等。
当捕捉/比较控制寄存器 00 (CRC00)的第 1 位=1 时，TM00 的计数值被捕捉到 CR000(在 TI000 引脚输入信号的反向沿捕捉)，但不产生捕捉中断信号(INTTM000)。然而当检测到 TI010 引脚的有效沿时产生 INTTM000 中断信号。
如果不使用 INTTM000 信号，则将其屏蔽。

**(4)　清零&启动模式(通过 TI000 引脚有效沿输入进入)下的操作**
**(CR000: 捕捉寄存器, CR010: 捕捉寄存器)**

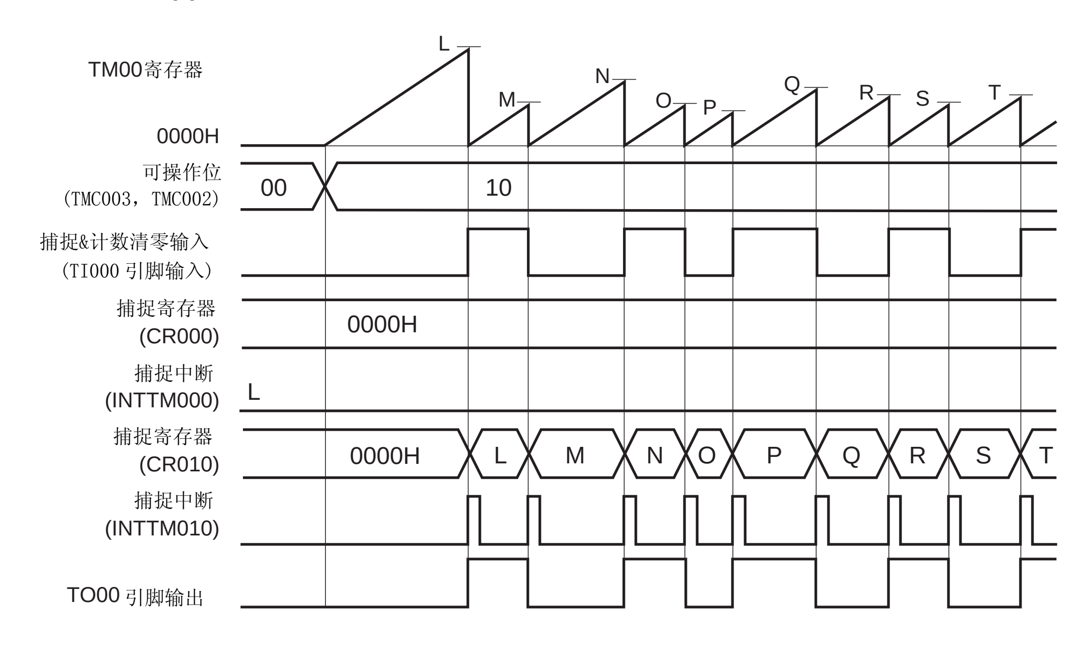
**图 6-28.　清零&启动模式(通过 TI000 引脚有效沿输入进入)的框图**
**(CR000: 捕捉寄存器, CR010: 捕捉寄存器)**




注　　当使用检测 TI010 引脚的有效沿时不能使用定时器输出(TO00)。

**图 6-29. 清零&启动模式(通过 TI000 引脚有效沿输入进入)时序示例**
**(CR000: 捕捉寄存器, CR010: 捕捉寄存器) (1/3)**

**(a) TOC00 = 13H, PRM00 = 30H, CRC00 = 05H, TMC00 = 0AH**



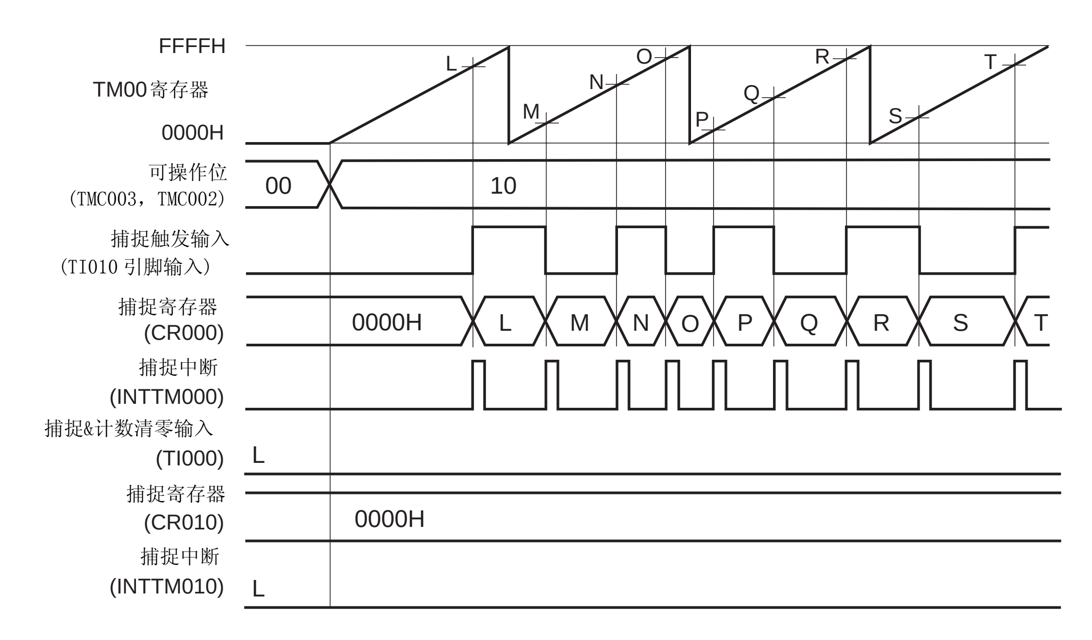

这是一个应用示例：当检测到 TI000 引脚的上升沿或下降沿时捕捉计数值存入 CR010、将 TM00 清零并反转 TO00 引脚的输出

在检测到 TI010 引脚的脉冲沿时产生中断信号(INTTM000)。如果不使用 INTTM000 信号，则将其屏蔽。

**图 6-29. 清零&启动模式(通过 TI000 引脚有效沿输入进入)时序示例 (CR000: 捕捉寄存器, CR010: 捕捉寄存器) (2/3)**

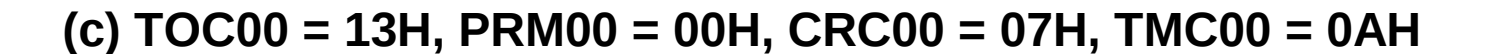


**(b) TOC00 = 13H, PRM00 = C0H, CRC00 = 05H, TMC00 = 0AH**

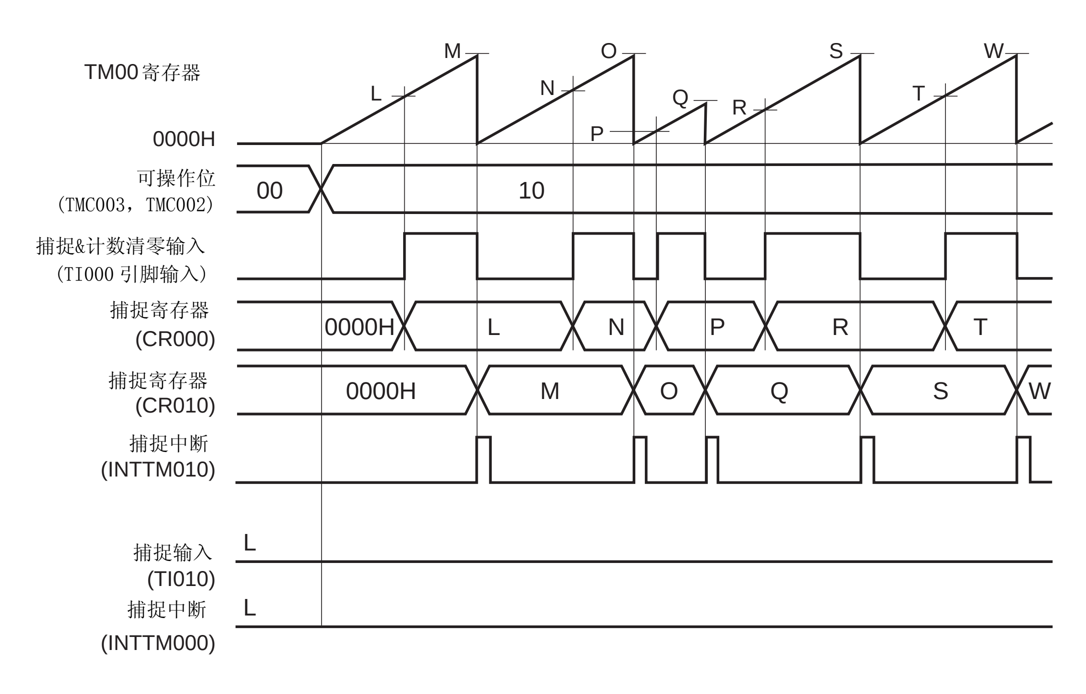



这是一个时序示例：当脉冲沿不输入到 TI000 引脚时，检测到 TI010 引脚的上升沿或下降沿时捕捉计数值存入 CR000 。

**图 6-29.　清零&启动模式(通过 TI000 引脚有效沿输入进入)时序示例**
**(CR000: 捕捉寄存器, CR010: 捕捉寄存器) (3/3)**



**(c) TOC00 = 13H, PRM00 = 00H, CRC00 = 07H, TMC00 = 0AH**



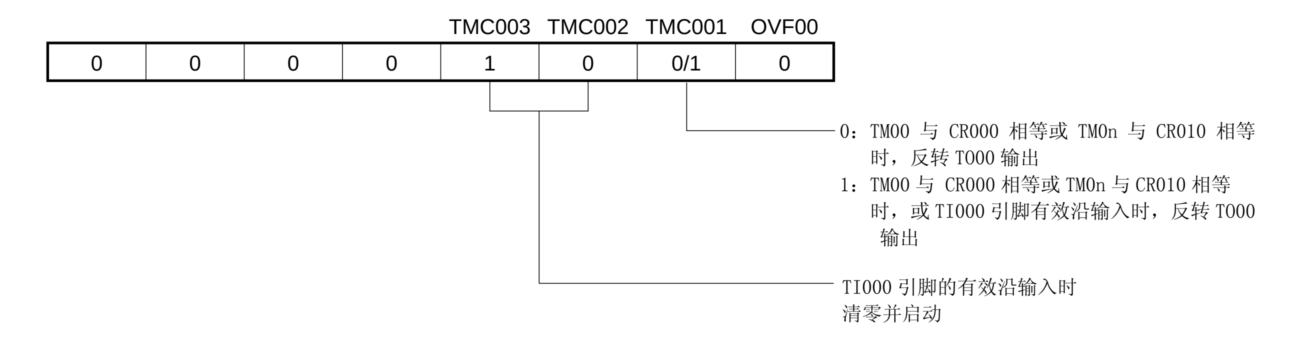

这是一个应用示例：对输入到 TI000 引脚的信号脉冲宽度进行测量。
通过设置 CRC00，在 TI000 引脚下降沿(也可以是上升沿)的反向沿捕捉计数值存入 CR000，而在 TI000 引脚下降沿存入 CR010。
可以通过下列表达式计算输入脉冲的高电平和低电平宽度。

- 高电平宽度 = [CR010 的值] – [CR000 的值] × [计数时钟周期]
- 低电平宽度 = [CR000 的值] × [计数时钟周期]

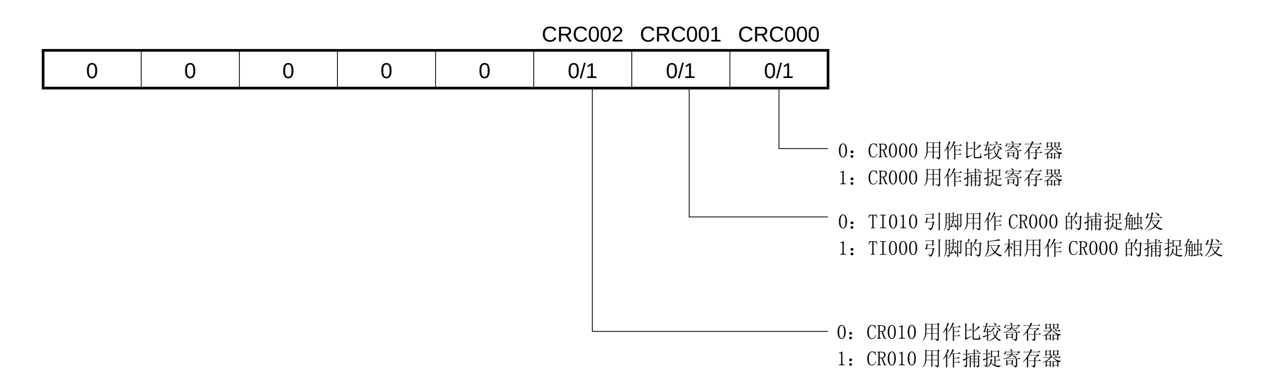
如果选择 TI000 引脚脉冲的反向沿作为触发，捕捉计数值并存入 CR000，则不会产生 INTTM000 信号。在产生 INTTM010 信号后立即读取 CR000 和 CR010，测量脉冲宽度。
但如果将预分频器模式寄存器 00 (PRM00)的第 6 和 5 位(ES101 和 ES100)指示的有效沿输入 TI010 引脚，则不会捕捉计数值，但会产生 INTTM000 信号。如果要测量 TI000 引脚的脉冲宽度，在不使用 INTTM000 信号时将其屏蔽。

**图 6-30. 在清零&启动模式(通过 TI000 引脚有效沿输入进入)下寄存器设置示例 (1/2)**

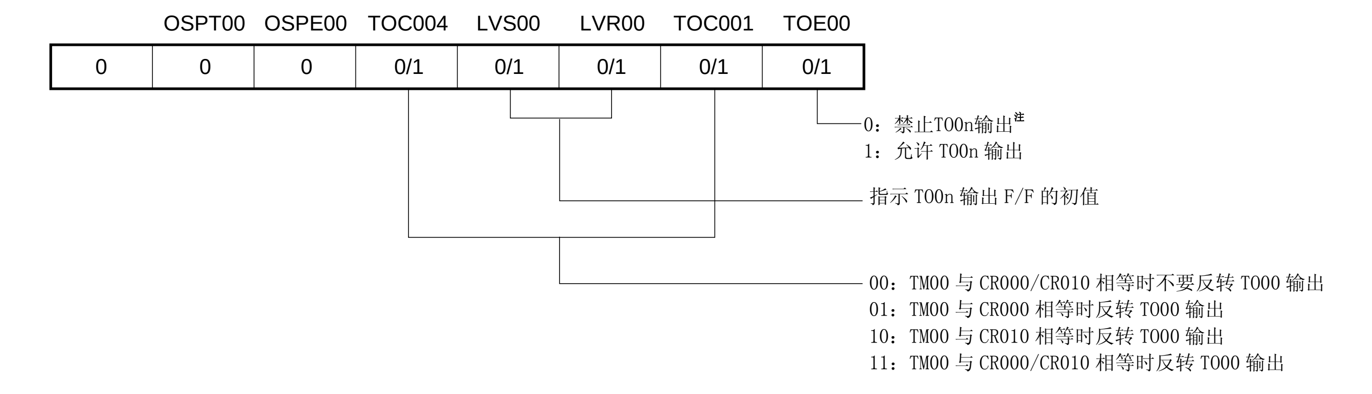
**(a) 16 位定时器模式控制寄存器 00 (TMC00)**




**(b) 捕捉/比较控制寄存器 00 (CRC00)**



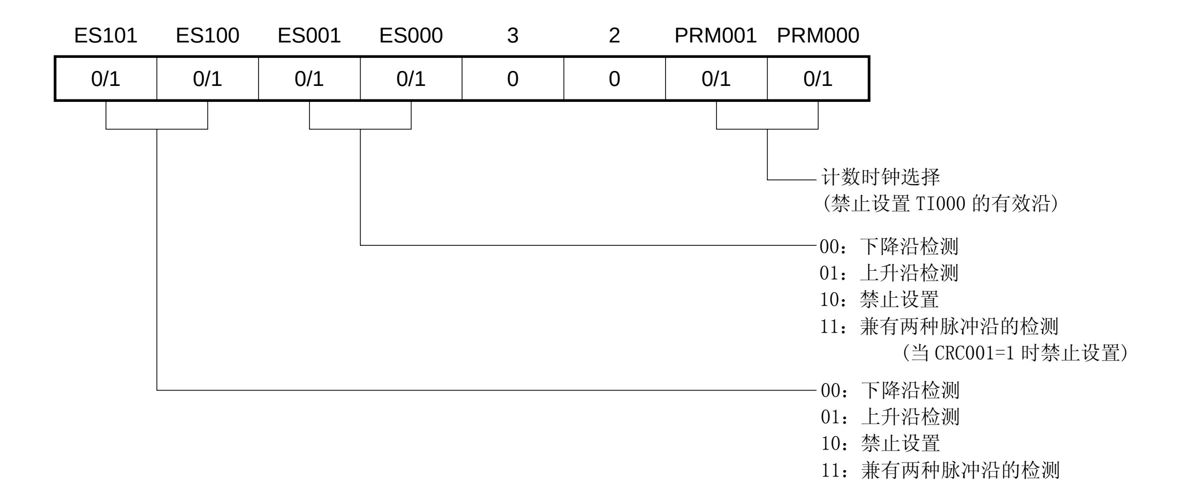

**(c) 16 位定时器输出控制寄存器 00 (TOC00)**




注　　当使用检测 TI010 引脚有效沿时不能使用定时器输出(TO00)。

**图 6-30. 在清零&启动模式(通过 TI000 引脚有效沿输入进入)下寄存器设置示例 (2/2)**

**(d) 预分频器模式寄存器 00 (PRM00)**




**(e) 16 位定时器计数器 00 (TM00)**

通过读取 TM00，获得计数值。

**(f) 16 位捕捉/比较寄存器 000 (CR000)**

当该寄存器用作比较寄存器并且当其值与 TM00 的值相等时，产生中断信号(INTTM000)。但不对 TM00 清零。

如果要将该寄存器用作捕捉寄存器，则应选择TI000 或TI010 引脚[注]输入作为捕捉触发。当检测到捕捉触发的有效沿时，将TM00 的计数值存入CR000。

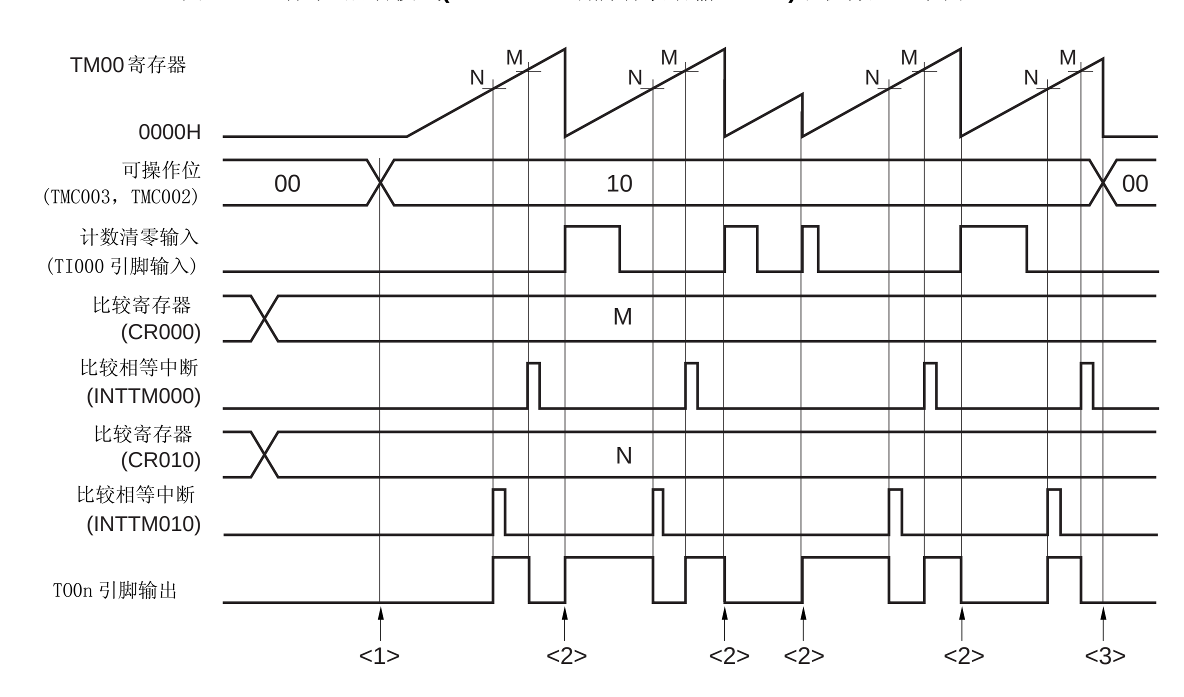
**注** 当使用检测 TI010 引脚的有效沿时不能使用定时器输出(TO00)。

**(g) 16 位捕捉/比较寄存器 010 (CR010)**

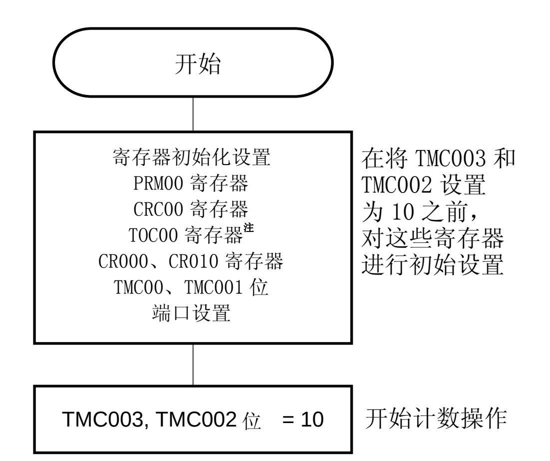
当该寄存器用作比较寄存器并且当其值与 TM00 的值相等时，产生中断信号(INTTM010)。但不对 TM00 清零。

当该寄存器用作捕捉寄存器，则 TI000 引脚输入作为捕捉触发。当检测到捕捉触发的有效沿时，将 TM00 的计数值存入 CR010。

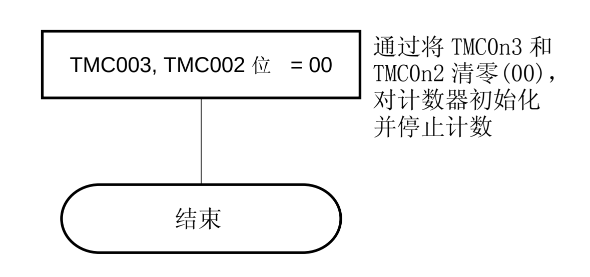
**图 6-31. 清零&启动模式(通过 TI000 引脚有效沿输入进入)下软件处理示例**



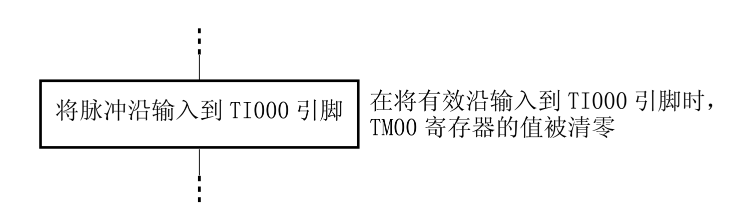

<1> 计数操作启动流程




<3> 计数操作停止流程




<2> TM00 寄存器清零&启动流程




注　设置 TOC00 时必须特别小心。如需了解详细信息，可参见 **6.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)**。

### 6.4.5 自由运行定时器操作

当 16 位定时器模式控制寄存器 00(TMC00)的第 3 和 2 位(TMC003 与 TMC002)为 01(自由运行定时器模式)时，16 位定时器/事件计数器 00 连续计数并与计数时钟同步。当计数值上升到 FFFFH 时，在下一个时钟将溢出标志(OVF00)设置为 1、将 TM00 清零(0000H)并继续计数。通过软件执行 CLR 指令将 OVF00 清零(0)。

可以使用以下三种自由运行定时器操作。

- CR000 和 CR010 都用作比较寄存器。
- CR000 与 CR010 一个用作比较寄存器，另一个用作捕捉寄存器。
- CR000 和 CR010 都用作捕捉寄存器。

备注　**1.** 如需了解 I/O 引脚的设置，参见 **6.3 (5) 端口模式寄存器 0(PM0)**。
　　　**2.** 要了解如何允许 INTTM000 信号中断，可参见 **第十八章 中断功能**。

**(1)** 自由运行定时器模式操作
**(CR000: 比较寄存器, CR010: 比较寄存器)**

图 **6-32.** 自由运行定时器模式的框图
**(CR000: 比较寄存器, CR010: 比较寄存器)**

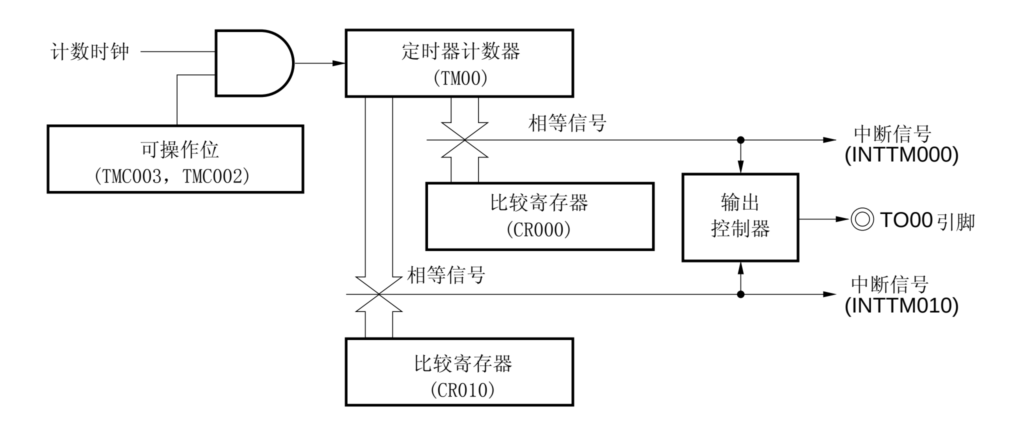
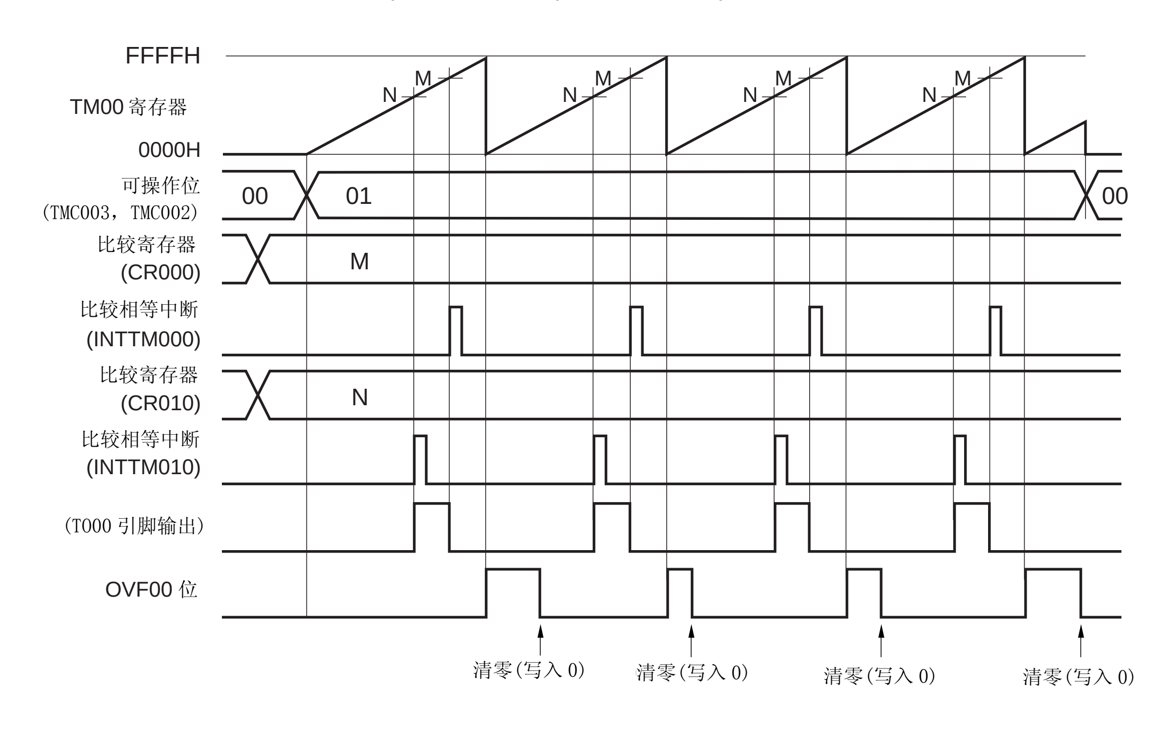


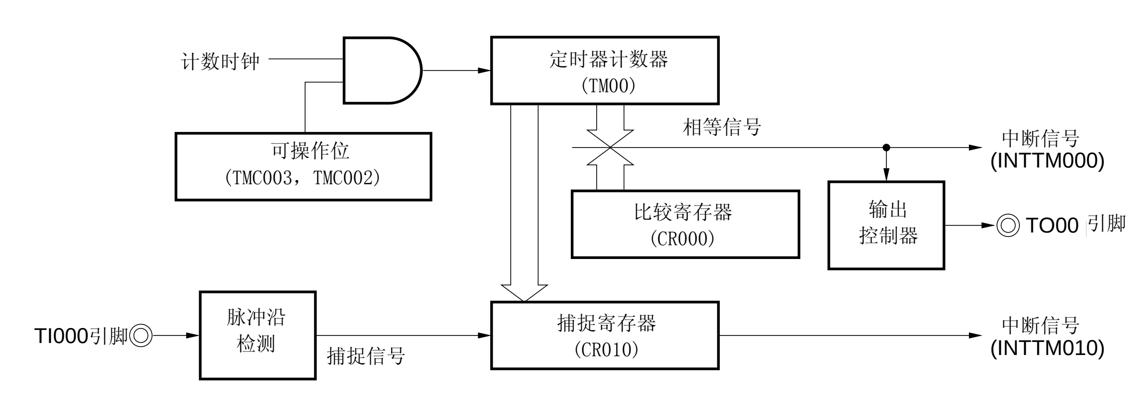
图 **6-33.** 自由运行定时器模式时序示例
**(CR000:** 比较寄存器**, CR010:** 比较寄存器**)**




这是一个应用示例：两个比较寄存器在自由运行定时器模式下使用。
每当 TM00 的计数值与 CR000 或 CR010 的值相等时，反转 TO00 引脚的输出电平。当计数值与寄存器值相等时，产生 INTTM000 或 INTTM010 信号。

**(2)** 自由运行定时器模式操作
**(CR000:** 比较寄存器**, CR010:** 捕捉寄存器**)**

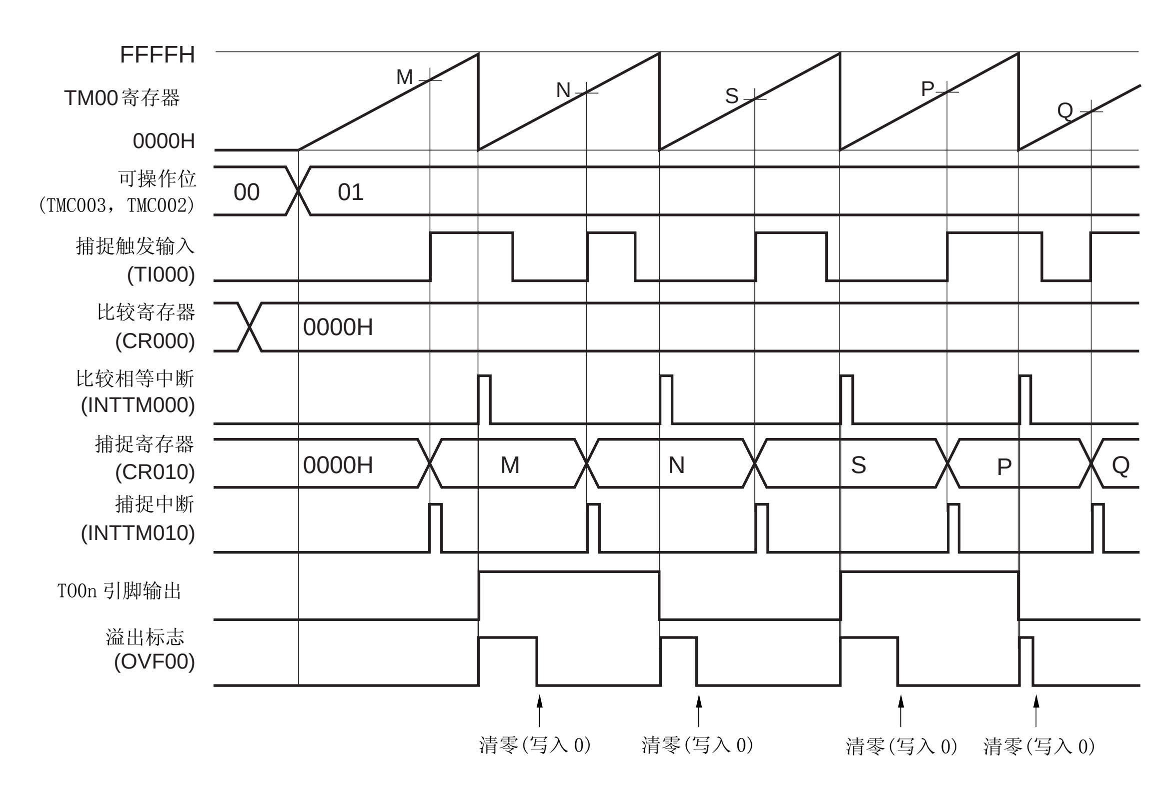
图 **6-34.** 自由运行定时器模式的框图
**(CR000:** 比较寄存器**, CR010:** 捕捉寄存器**)**



<R>

**图 6-35. 自由运行定时器模式时序示例**
**(CR000: 比较寄存器, CR010: 捕捉寄存器)**

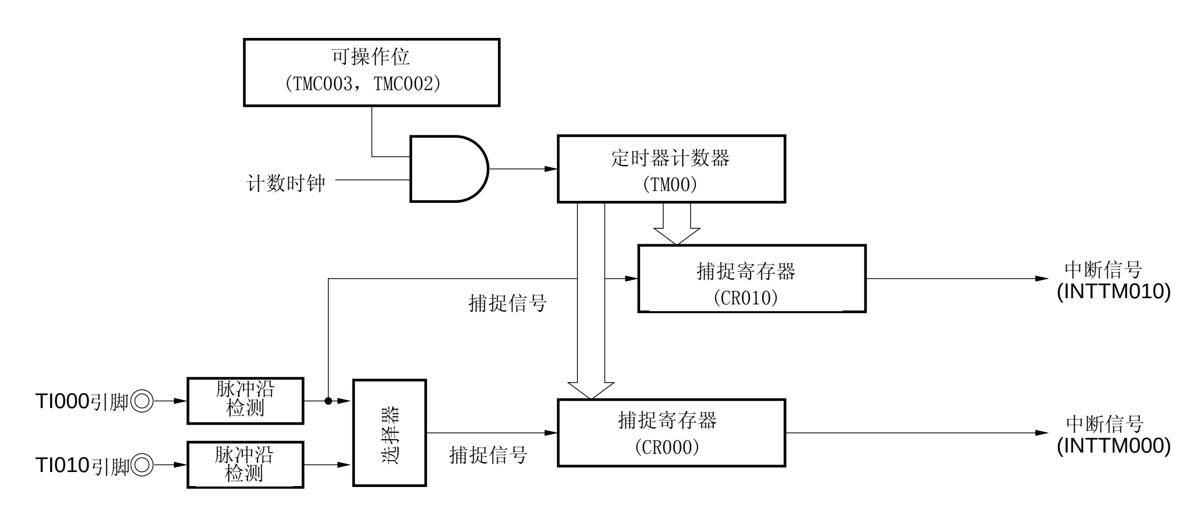
**• TOC00 = 13H, PRM00 = 10H, CRC00 = 04H, TMC00 = 04H**




这是一个应用示例：在自由运行定时器模式下同时使用一个比较寄存器和一个捕捉寄存器。
在本例中每当 TM00 的计数值与 CR000 的值相等时反转 TO00 引脚的输出电平并且产生中断 INTTM000。此外每当检测到 TI000 引脚的有效沿时，产生 INTTM010 信号，并捕捉 TM00 的计数值存入 CR010。

**(3)** 自由运行定时器模式操作
**(CR000: 捕捉寄存器, CR010: 捕捉寄存器)**

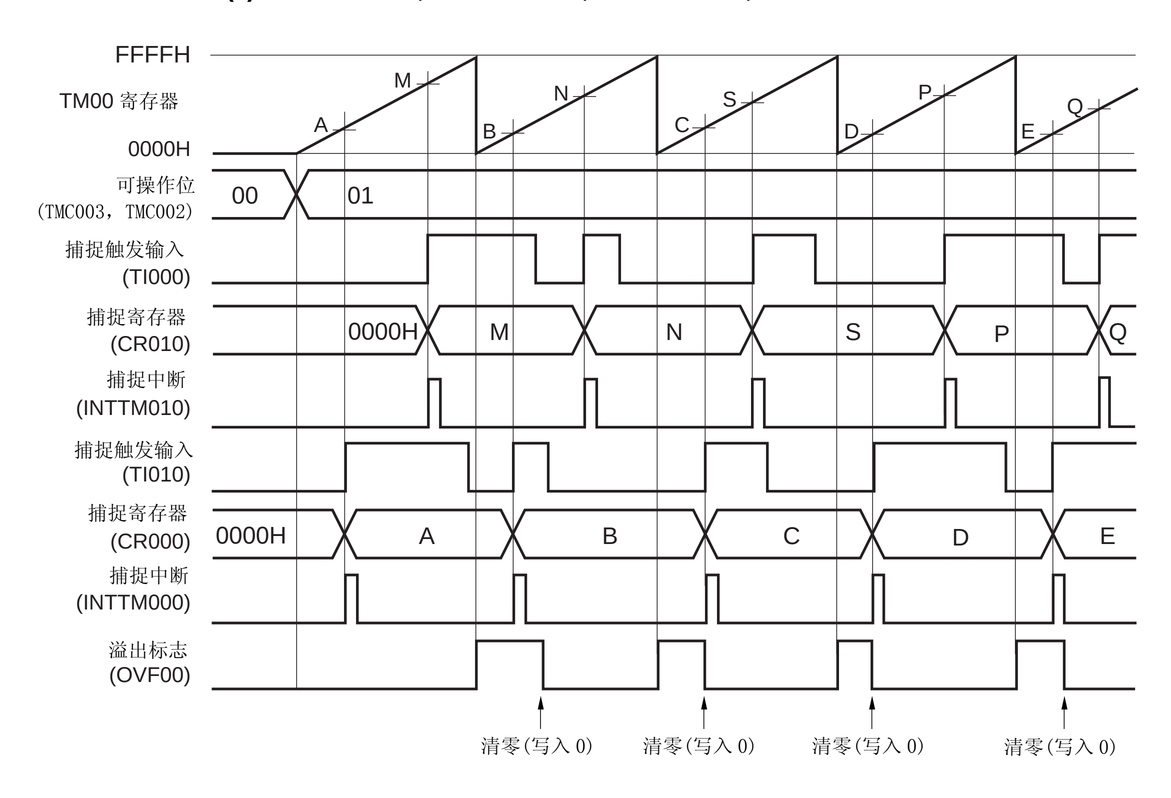
图 **6-36.** 自由运行定时器模式的框图
**(CR000: 捕捉寄存器, CR010: 捕捉寄存器)**




**备注** 如果在自由运行定时器模式下，CR000 与 CR010 用作捕捉寄存器，则不反转 TO00 引脚的输出电平。但如果 16 位定时器模式控制寄存器 00 (TMC00)的第 1 位(TMC001)=1，则每次检测到 TI000 引脚的有效沿时可以反转。

图 **6-37.** 自由运行定时器模式时序示例
**(CR000: 捕捉寄存器, CR010: 捕捉寄存器) (1/2)**

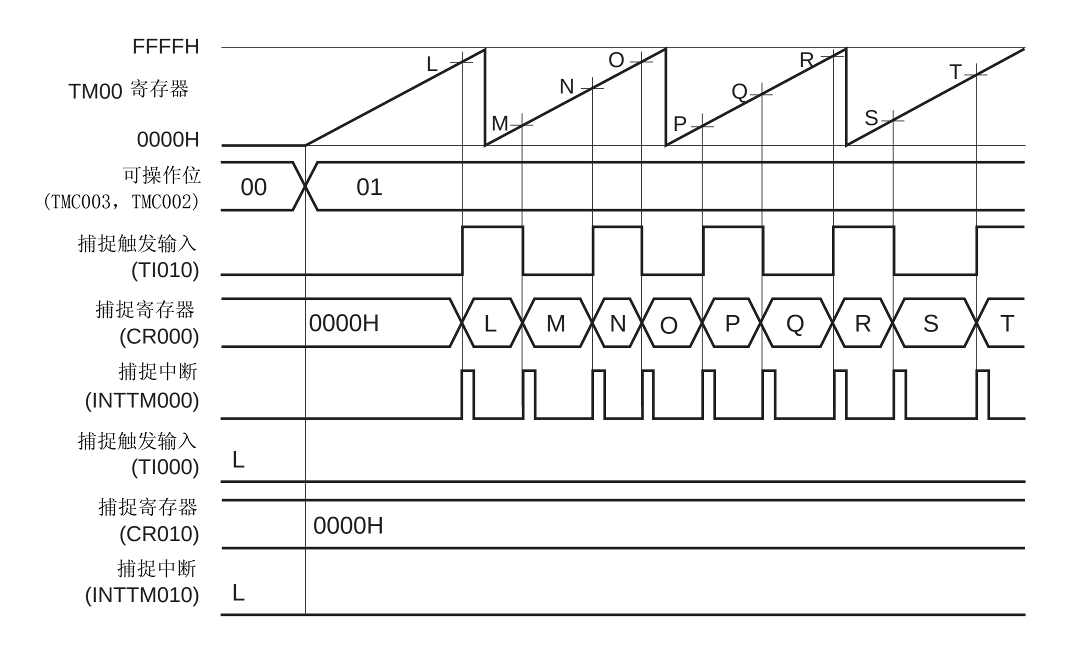
**(a) TOC00 = 13H, PRM00 = 50H, CRC00 = 05H, TMC00 = 04H**




这是一个应用示例：自由运行定时器模式下，在各自触发信号的有效沿捕捉的计数值存入各自的捕捉寄存器中。
当检测到 TI000 引脚输入的有效沿时捕捉计数值并存入 CR010，而当检测到 TI010 引脚输入的有效沿时捕捉计数值并存入 CR000。

**图 6-37.  自由运行定时器模式时序示例**
**(CR000: 捕捉寄存器, CR010: 捕捉寄存器) (2/2)**

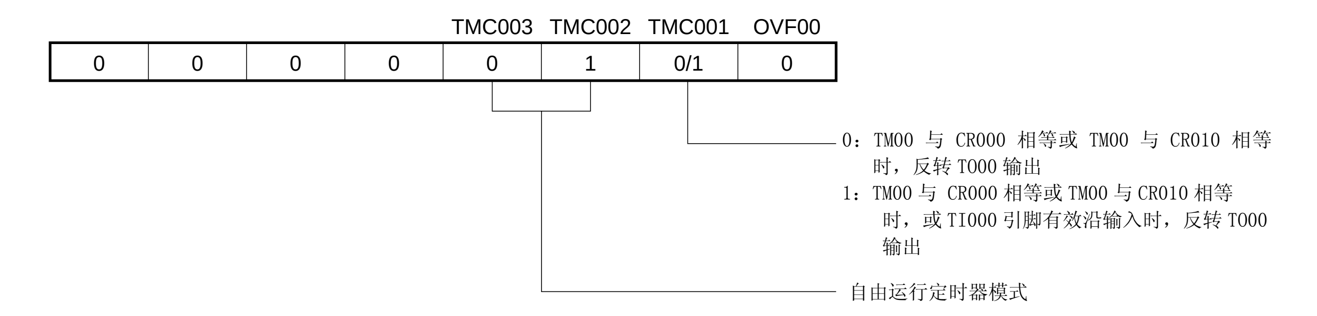
**(b)  TOC00 = 13H, PRM00 = C0H, CRC00 = 05H, TMC00 = 04H**



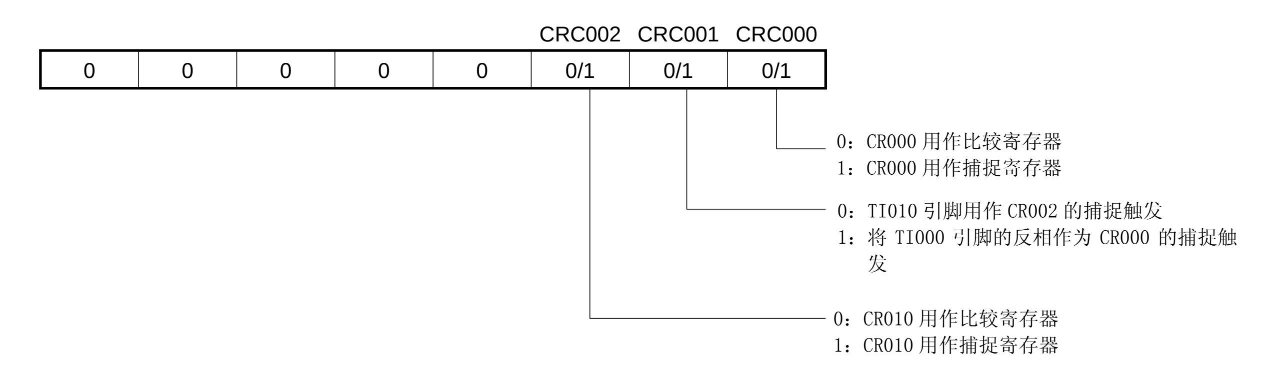

这是一个应用示例：自由运行定时器模式下，检测 TI010 引脚的两种脉冲沿，计数值被捕捉到 CR000。
当 CR000 与 CR010 都用作捕捉寄存器并且仅检测到 TI010 引脚的有效沿时，计数值不能被捕捉到 CR010。

图 6-38. 自由运行定时器模式下寄存器设置示例 (1/2)

**(a) 16 位定时器模式控制寄存器 00 (TMC00)**




**(b) 捕捉/比较控制寄存器 00 (CRC00)**




**(c) 16 位定时器输出控制寄存器 00 (TOC00)**

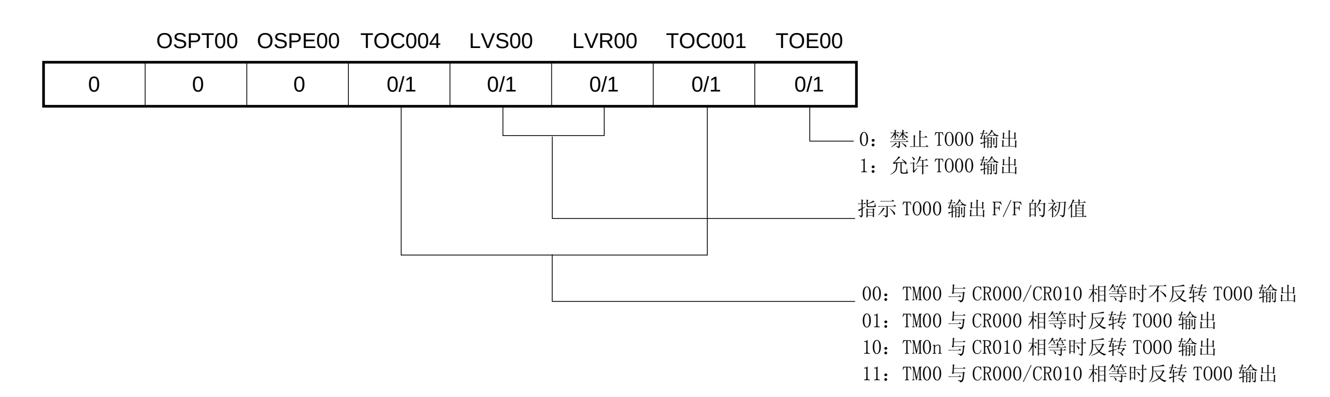
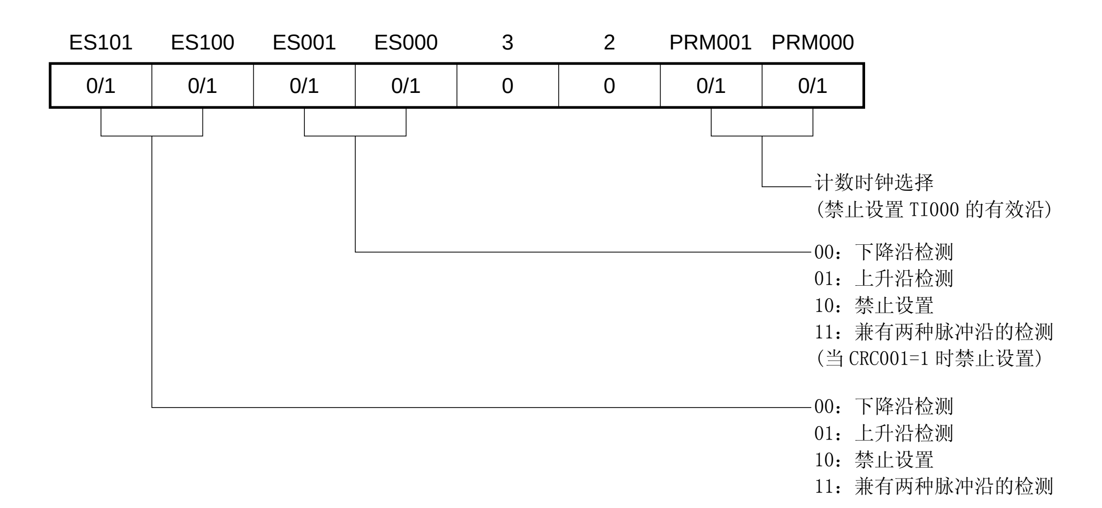


图 6-38. 自由运行定时器模式下寄存器设置示例 (2/2)

**(d) 预分频器模式寄存器 00 (PRM00)**




**(e) 16 位定时器计数器 00 (TM00)**

通过读取 TM00，获得计数值。

**(f) 16 位捕捉/比较寄存器 000 (CR000)**

当该寄存器用作比较寄存器并且当其值与 TM00 的计数值相等时，产生中断信号(INTTM000)。但不对 TM00 计数值清零。

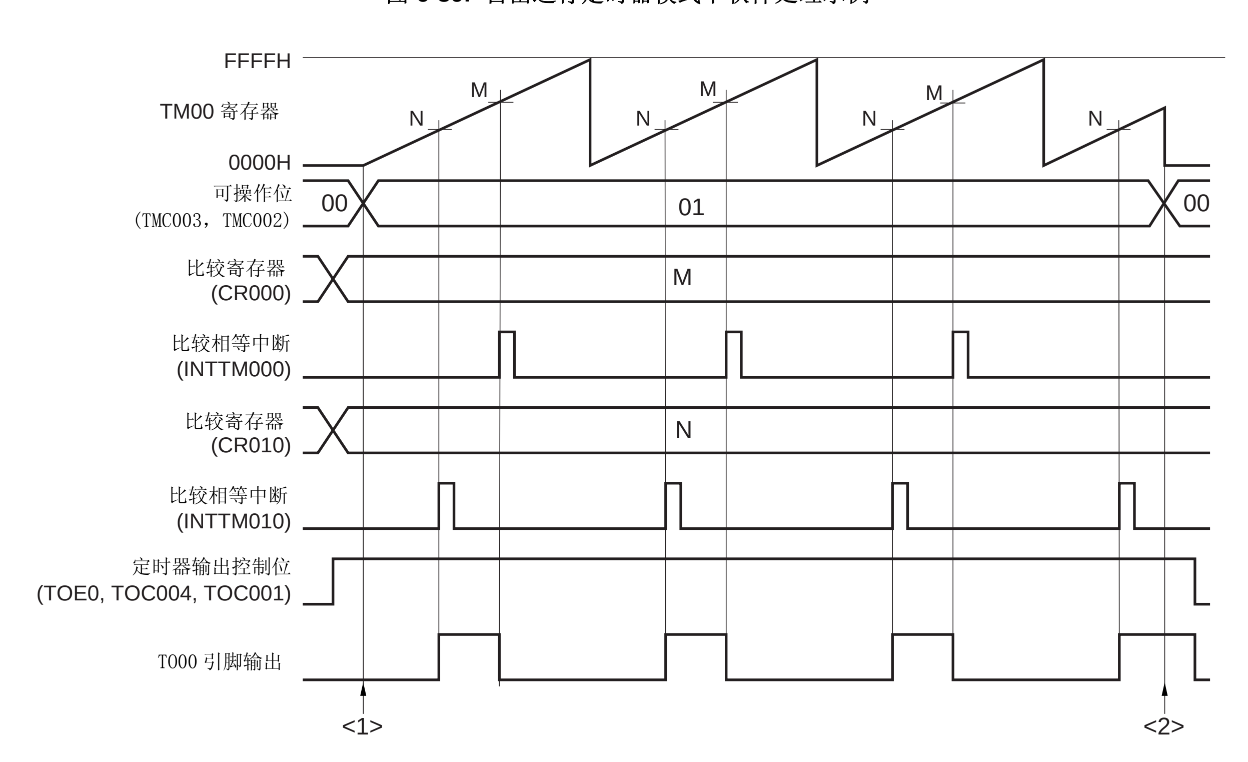
若要用作捕捉寄存器，则应选择 TI000 或 TI010 引脚输入作为捕捉触发。当检测到捕捉触发的有效沿时，将 TM00 的计数值存入 CR000。

**(g) 16 位捕捉/比较寄存器 010 (CR010)**

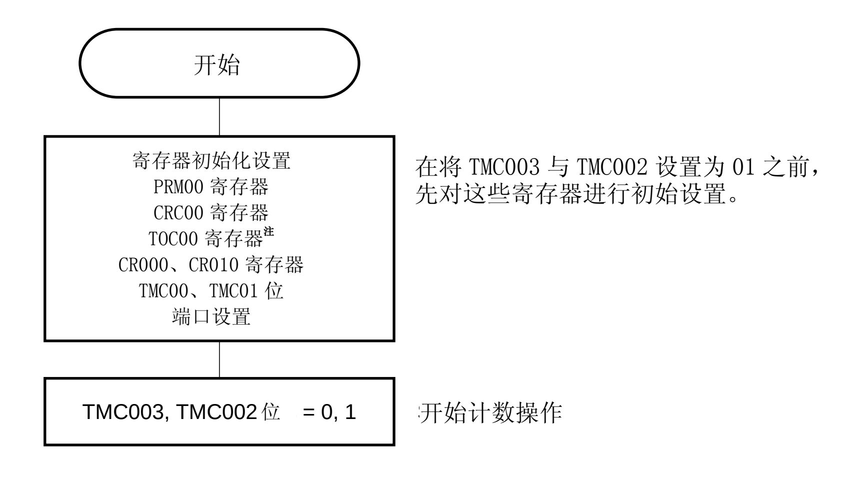
当该寄存器用作比较寄存器并且当其值与 TM00 的计数值相等时，产生中断信号(INTTM010)。但不对 TM00 计数值清零。

当该寄存器用作捕捉寄存器时，将 TI000 引脚输入作为捕捉触发。当检测到捕捉触发的有效沿时，将 TM00 的计数值存入 CR010。

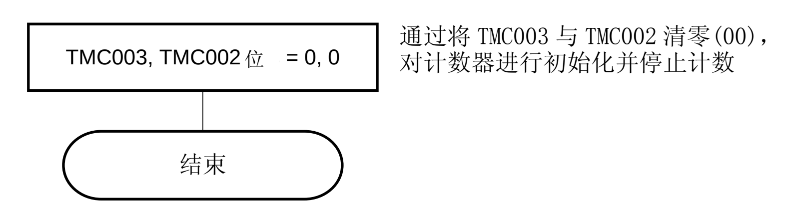
图 6-39. 自由运行定时器模式下软件处理示例




<1> 计数操作启动流程




<2> 计数操作停止流程




注　要特别注意 TOC00 的设置。如需了解详细信息，可参见 **6.3 (3) 16 位定时器输出控制寄存器 00 (TOC00)。**

### 6.4.6 PPG 输出操作

当 16 位定时器模式控制寄存器 00 (TMC00) 的第 3 和 2 位(TMC003 与 TMC002)=11 时(在 TM00 与 CR000 相等时清零&启动)，在由 CR000 设置的周期内从 TO00 引脚输出方波(具有通过 CR010 预先设置的一个脉冲宽度)作为 PPG(Programmable Pulse Generator 可编程脉冲发生器)信号。

以下是作为 PPG 输出的脉冲的周期与占空比。

- 脉冲周期 = (CR000 的设置值 + 1) × 计数时钟周期
- 占空比 = (CR010 的设置值 + 1) / (CR000 的设置值 + 1)

**注意事项　若在操作期间修改占空比（CR010 寄存器的值），可参阅 6.5.1 TM00 操作期间 CR010 的重写。**

备注　1. 如需了解 I/O 引脚的设置，可参见 **6.3 (5) 端口模式寄存器 0 (PM0)**。

2. 要了解如何允许 INTTM000 信号中断，参见 **第十八章 中断功能**。

**图 6-40. PPG 输出操作的框图**

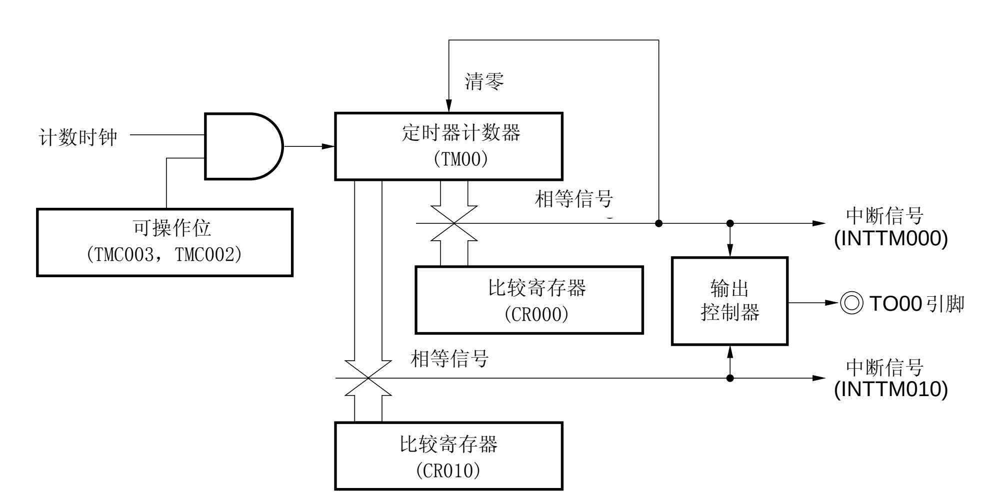
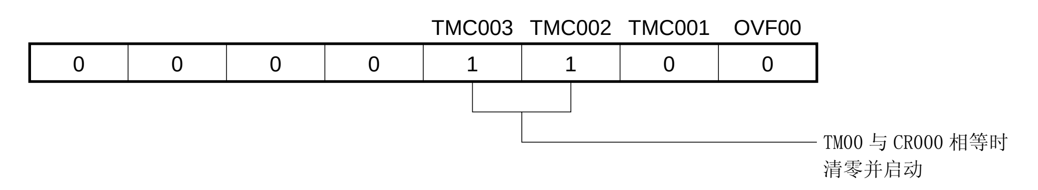
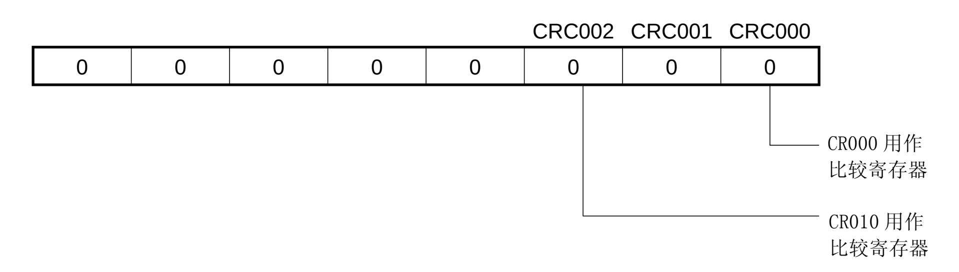


图 6-41. PPG 输出操作的寄存器设置示例

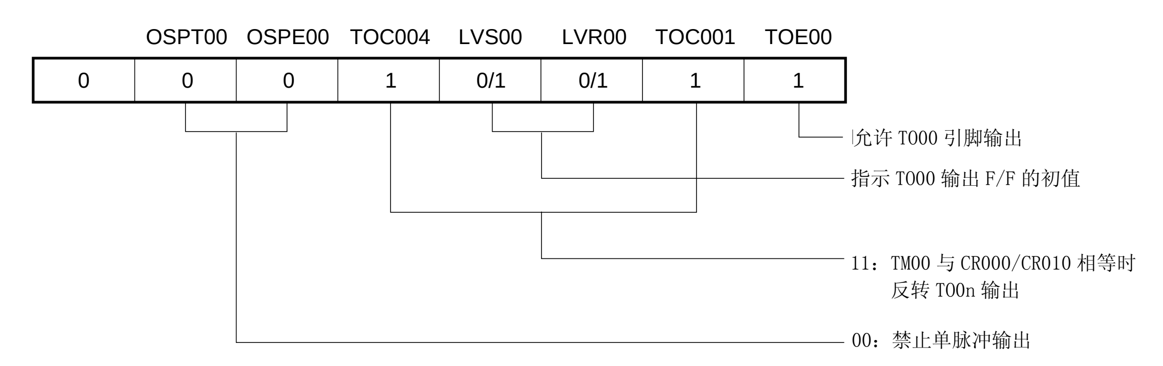
**(a) 16 位定时器模式控制寄存器 00 (TMC00)**



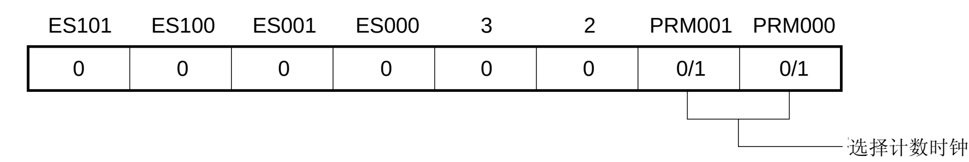

**(b) 捕捉/比较控制寄存器 00 (CRC00)**




**(c) 16 位定时器输出控制寄存器 00 (TOC00)**




**(d) 预分频器模式寄存器 00 (PRM00)**




**(e) 16 位定时器计数器 00 (TM00)**

通过读取 TM00，获得计数值。

**(f) 16 位捕捉/比较寄存器 000 (CR000)**

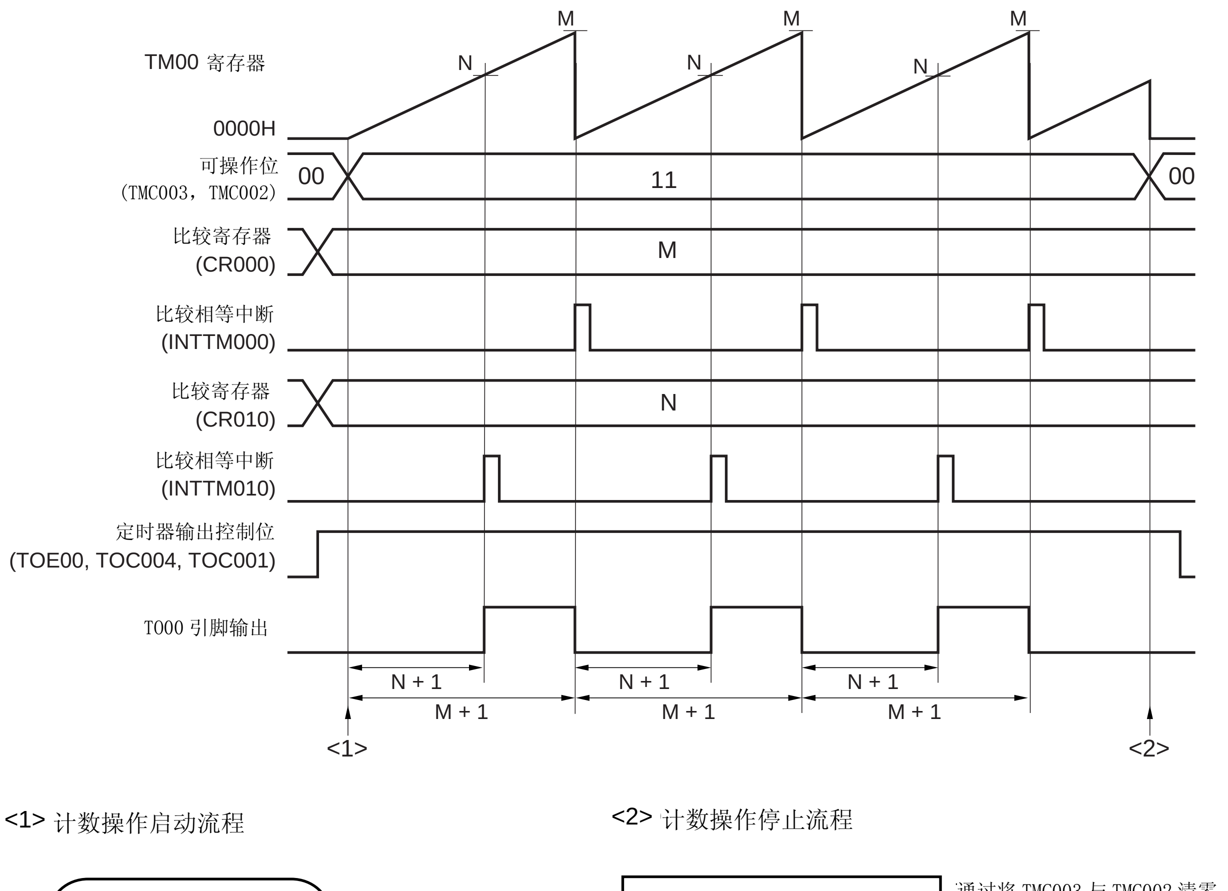
当该寄存器的值与 TM00 的计数值相等时，产生中断信号(INTTM000)。但不对 TM00 的计数值清零。

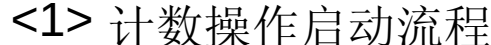
**(g) 16 位捕捉/比较寄存器 010 (CR010)**

当该寄存器的值与 TM00 的计数值相等时，产生中断信号(INTTM010)。但不对 TM00 的计数值清零。

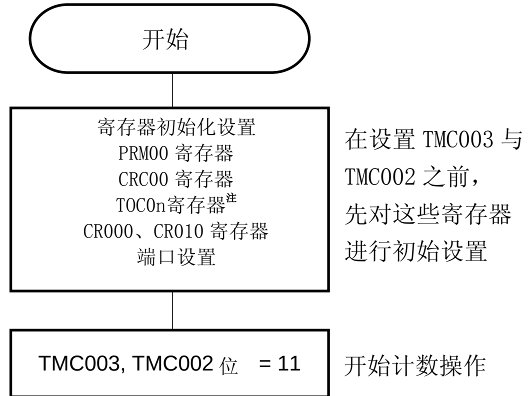
<R> **注意事项** 对 **CR000** 与 **CR010** 进行设置，以满足条件 **0000H ≤ CR010 < CR000 ≤ FFFFH** 。

图 6-42.  PPG 输出操作软件处理示例

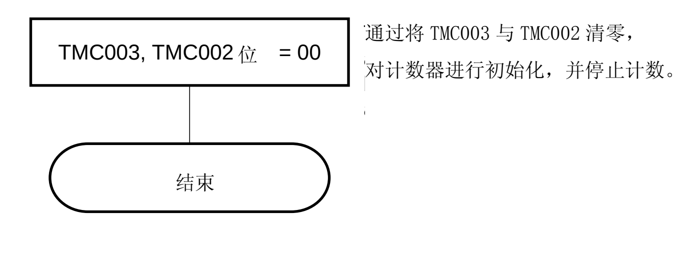


<1> 计数操作启动流程




<2> 计数操作停止流程




**注**　必须特别注意 TOC00 的设置。如需了解详细信息，可参见 **6.3 (3) 16 位定时器输出控制寄存器 00(TOC00)**。

**备注**　PPG 脉冲周期 = (M + 1) × 计数时钟周期
　　　　PPG 占空比 = (N + 1)/(M + 1)

### 6.4.7 单脉冲输出操作

通过将 16 位定时器模式控制寄存器 00 (TMC00) 的第 3 和 2 位(TMC003 与 TMC002)设置为 01 (自由运行定时器模式)或 10(清零&启动模式 (通过 TI000 引脚有效沿进入))，以及将 16 位定时器输出控制寄存器 00 (TOC00)的第 5 位(OSPE00)设置为 1，可以输出一个单脉冲。

当 TOC00 的第 6 位(OSPT00)=1 时或者在定时器操作期间将有效沿输入到 TI000 引脚时，触发 TM00 的清零&启动，并且从 TO00 引脚只输出一次反映 CR000 与 CR010 之间差距的脉冲。

**注意事项　1. 当单脉冲输出时，不要再次输入触发(设置 OSPT00=1 或者检测 TI000 引脚的有效沿)。在当前单脉冲输出完成后产生触发，才可再次输出单脉冲。**

**2. 如果要使用单脉冲输出触发(仅设置 OSPT00=1)，则不要修改 TI000 引脚或其复用功能端口引脚的电平。否则可能会产生异常输出的脉冲。**

备注　**1.** 如需了解 I/O 引脚的设置，可参见 **6.3 (5) 端口模式寄存器 0 (PM0)**.

**2.** 要了解如何允许 INTTM000 信号中断，可参见 **第十八章 中断功能**。

图 **6-43.** 单脉冲输出操作的框图

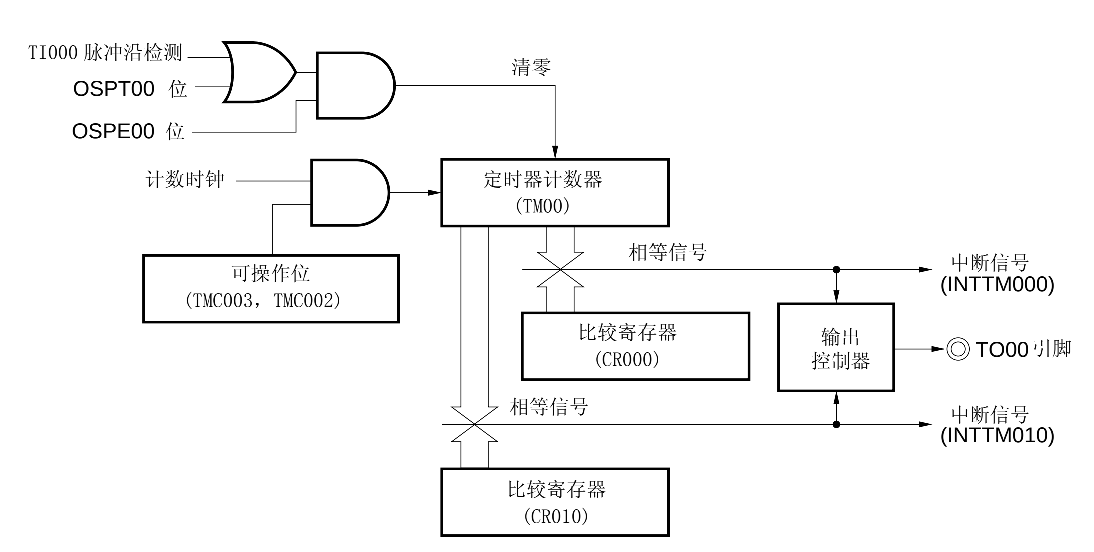
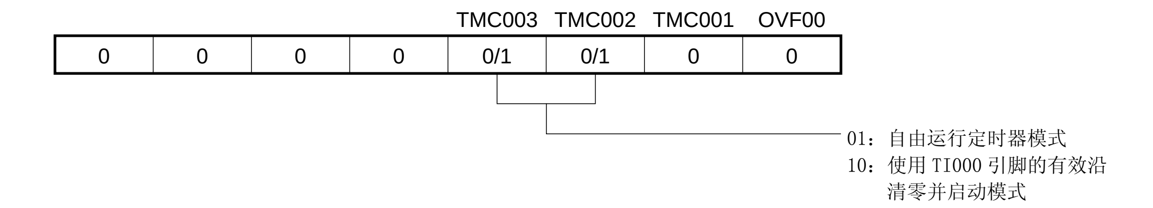
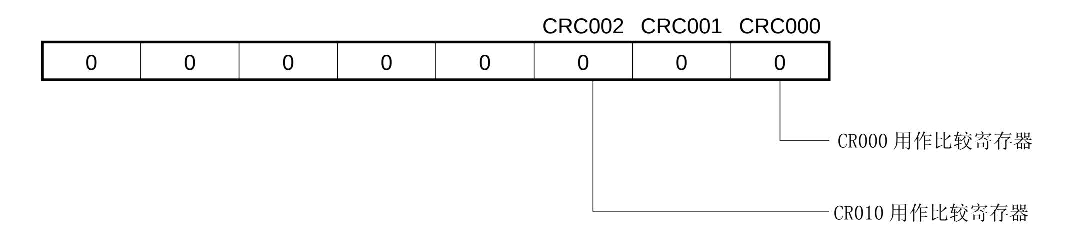


图 6-44. 单脉冲输出操作的寄存器设置示例 (1/2)

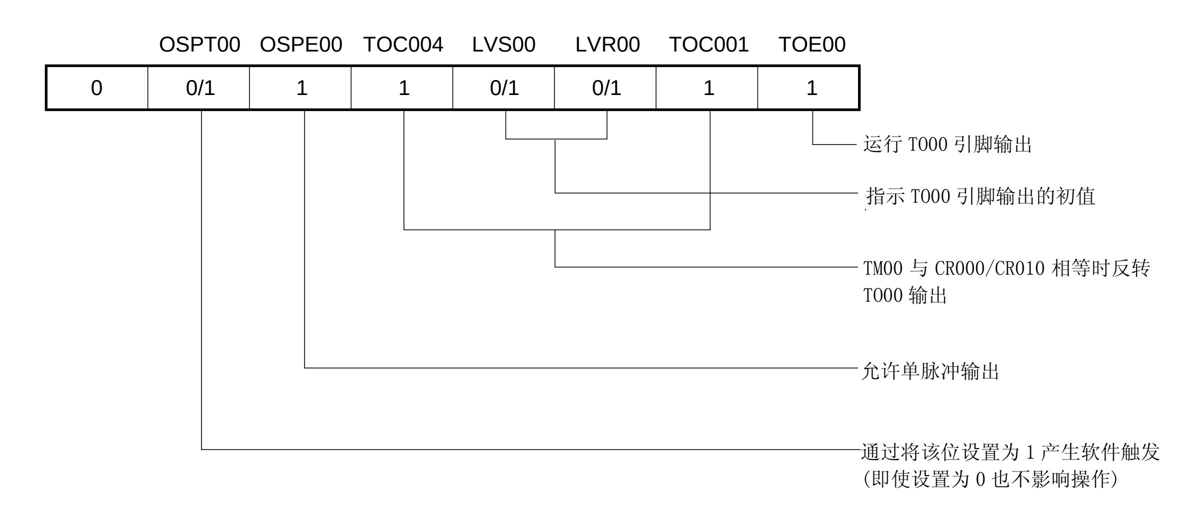
(a) 16 位定时器模式控制寄存器 00 (TMC00)




(b) 捕捉/比较控制寄存器 00 (CRC00)




(c) 16 位定时器输出控制寄存器 00 (TOC00)




(d) 预分频器模式寄存器 00 (PRM00)

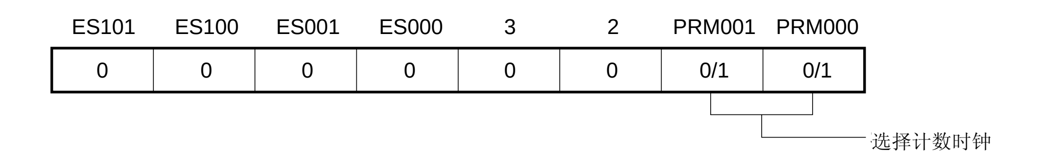


图 **6-44.** 单脉冲输出操作的寄存器设置示例**(2/2)**

**(e) 16 位定时器计数器 00 (TM00)**

通过读取 TM00，可获得计数值。

**(f) 16 位捕捉/比较寄存器 000 (CR000)**

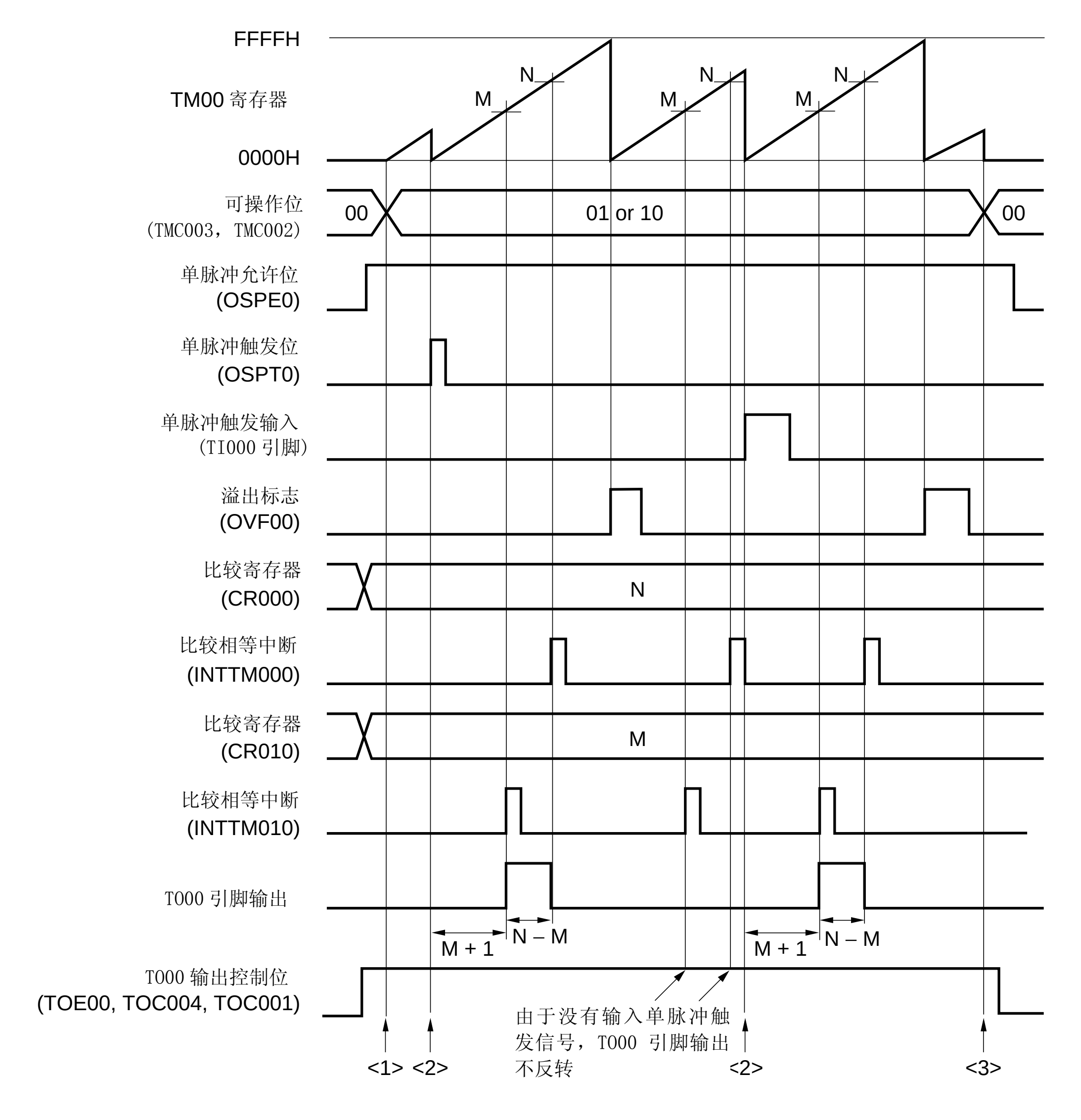
当输出单脉冲时该寄存器用作比较寄存器。当 TM00 的值与 CR000 的值相等时，产生中断信号(INTTM000)并反转 TO00 引脚的输出电平。

**(g) 16 位捕捉/比较寄存器 010 (CR010)**

当输出单脉冲时，该寄存器用作比较寄存器。当 TM00 的值与 CR010 的值相等时，产生中断信号(INTTM010)并反转 TO00 引脚的输出电平。

<R> 注意事项 不要为 **CR000** 与 **CR010** 设置相同的值。

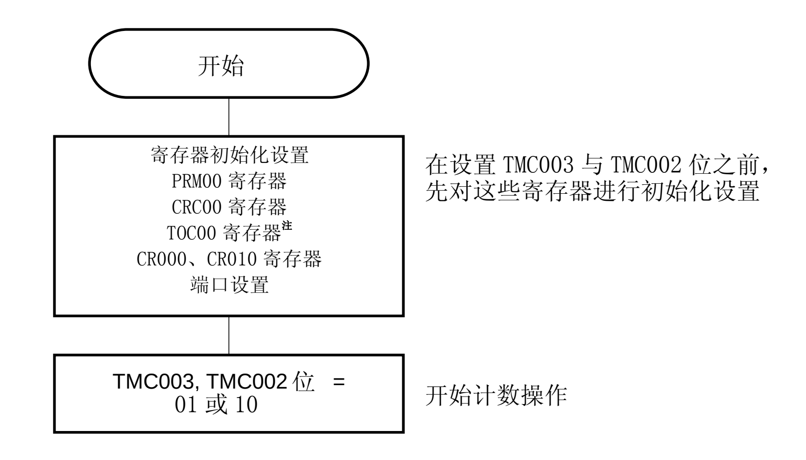
**图 6-45. 单脉冲输出操作软件处理示例 (1/2)**



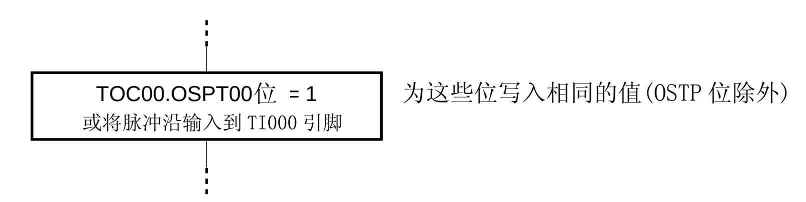

- 从输入单脉冲触发信号到输出单脉冲的时间
  = (M + 1) × 计数时钟周期
- 单脉冲输出活动电平宽度
  = (N – M) ×计数时钟周期

**图 6-45. 单脉冲输出操作软件处理示例 (2/2)**

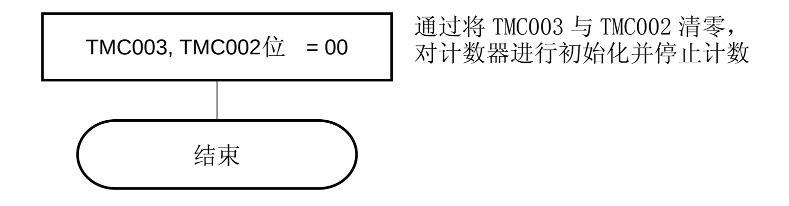
<1> 计数操作启动流程




<2> 单脉冲触发输入流程




<3> 计数操作停止流程




**注** 必须特别注意 TOC00 的设置。如需了解详细信息，可参见 **6.3 (3) 16 位定时器输出控制寄存器 00(TOC00)**。

### 6.4.8 脉冲宽度测量操作

TM00 可用于测量输入到 TI000 与 TI010 引脚的信号的脉冲宽度。

可以通过在自由运行定时器模式下操作 16 位定时器/事件计数器 00 以及通过同步使用 TI000 引脚输入信号的脉冲沿重新启动定时器，来进行测量。

当产生中断时，读取捕捉寄存器的有效值，并测量脉冲宽度。检测 16 位定时器模式控制寄存器 00 (TMC00)的第 0 位(OVF00)。如果该位被设置为 1，则由软件将其清零。

**图 6-46. 脉冲宽度测量 (自由运行定时器模式) 的框图**

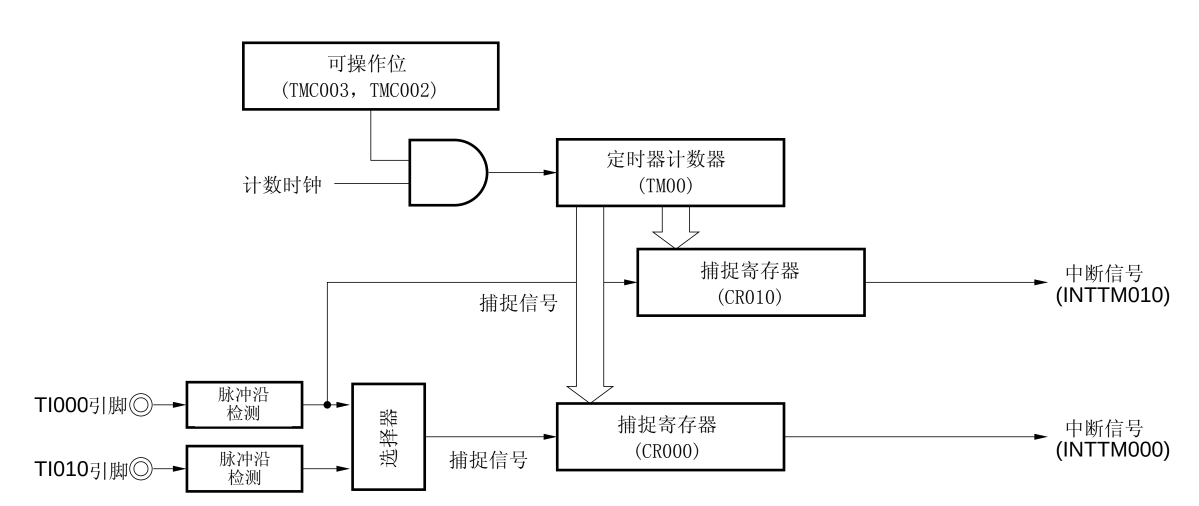
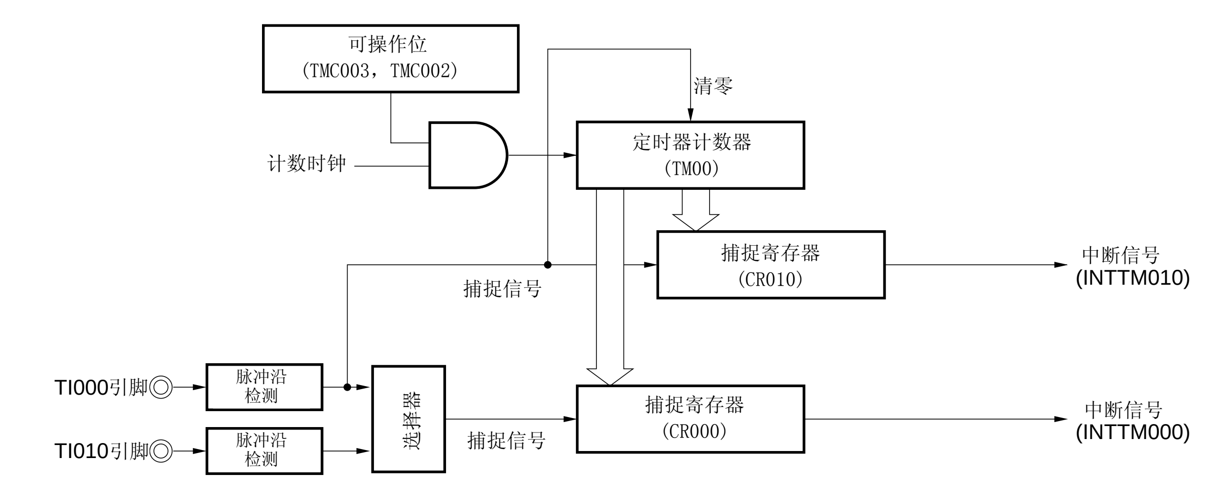



**图 6-47. 脉冲宽度测量的框图**
**(清零&启动模式(通过 TI000 引脚有效沿输入进入))**



可以使用以下三种方法测量脉冲宽度。

- 通过使用 TI000 与 TI010 引脚的两个输入信号测量脉冲宽度(自由运行定时器模式)
- 通过使用 TI000 引脚的一个输入信号测量脉冲宽度(自由运行定时器模式)
- 通过使用 TI000 引脚的一个输入信号测量脉冲宽度(清零&启动模式(通过 TI000 引脚有效沿输入进入))

备注 **1.** 如需了解 I/O 引脚的设置，可参见 **6.3 (5) 端口模式寄存器 0(PM0)**。
**2.** 要了解如何允许 INTTM000 信号中断，可参见 **第十八章 中断功能**。

**(1) 通过使用 TI000 与 TI010 引脚的两个输入信号测量脉冲宽度 (自由运行定时器模式)**

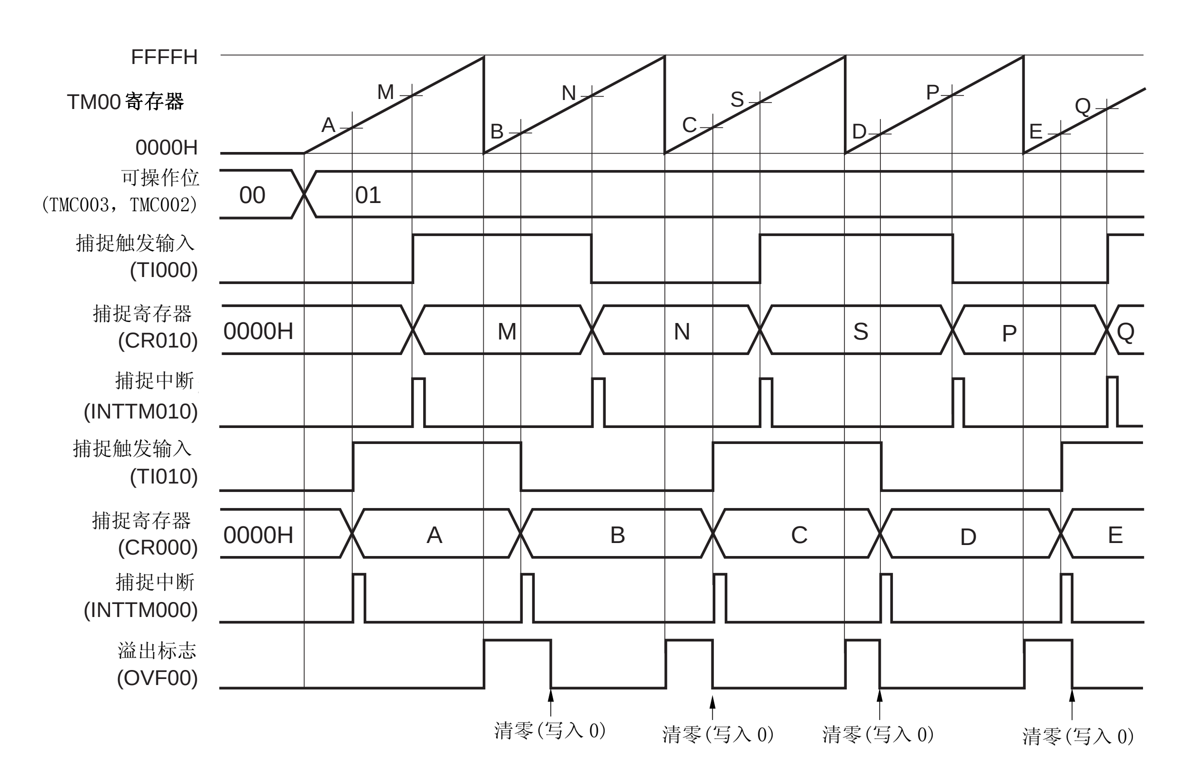
设置自由运行定时器模式(TMC003 与 TMC002 = 01)。当检测到 TI000 引脚的有效沿时，TM00 的计数值被捕捉到 CR010。当检测到 TI010 引脚的有效沿时，TM00 的计数值被捕捉到 CR000。规定检测 TI000 与 TI010 引脚的两种脉冲沿。

使用这种测量方法，要从每个输入信号的脉冲沿捕捉的计数值中减去预设值。因此要将先前捕捉的值事先存入一个单独的寄存器中。

如果产生溢出，则在当前捕捉值减去先前捕捉的值后计数值为负，并因此产生借位(将程序状态字(PSW)的第 0 位(CY)置 1)。如果发生这种情况，可以忽略 CY 并将计算结果作为脉冲宽度。此外，将 16 位定时器模式控制寄存器 00 (TMC0n)的第 0 位(OVF00)清零(0)。

图 **6-48.** 脉冲宽度测量时序示例 **(1)**

**• TMC00 = 04H, PRM00 = F0H, CRC00 = 05H**



**(2) 通过使用 TI00n 引脚的一个输入信号测量脉冲宽度(自由运行模式)**

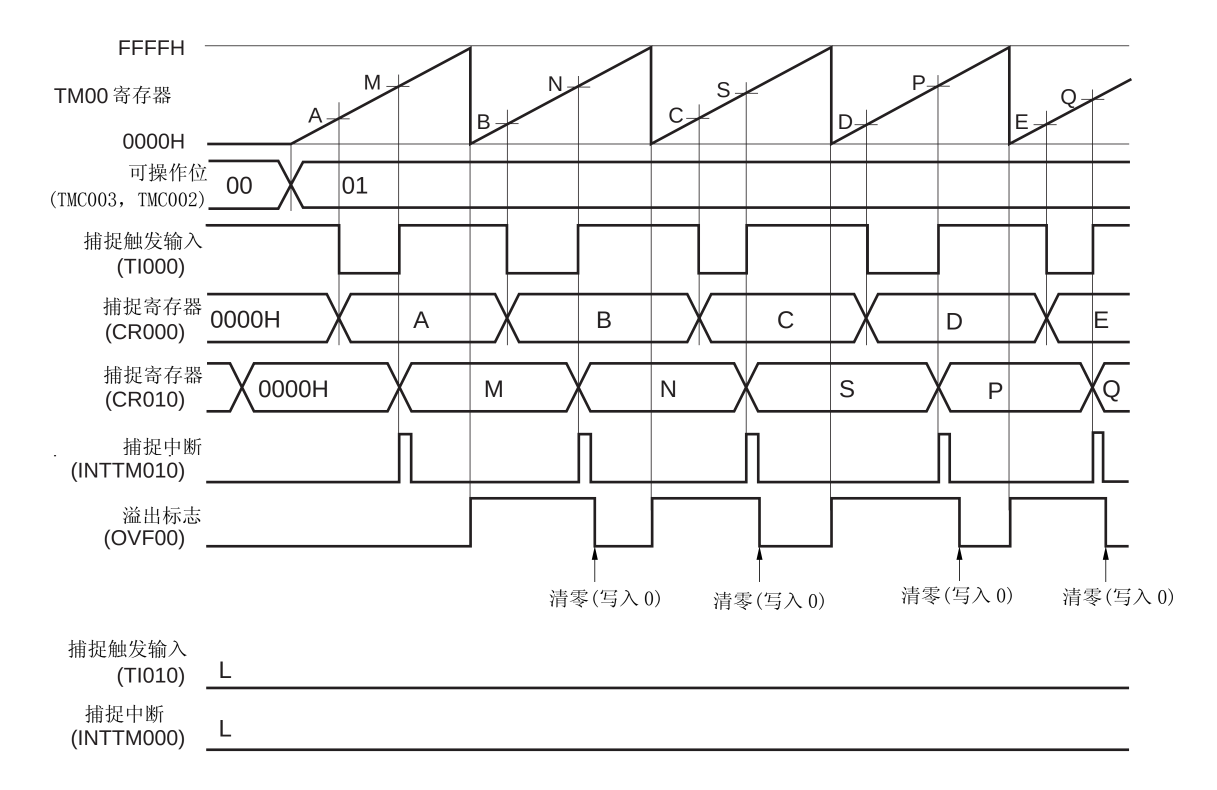
设置自由运行定时器模式(TMC003 与 TMC002 = 01)。在 TI000 引脚检测到反向的有效沿时，将 TM00 的计数值捕捉到 CR000。当检测到 TI000 引脚的有效沿时，将 TM00 的计数值捕捉到 CR010。

使用这种测量方法，在测量从一个脉冲沿到另一个脉冲沿的宽度时将数值存入分开的两个捕捉寄存器中。因此，不必保存捕捉值。将两个捕捉寄存器的值相减，可以计算高电平宽度、低电平宽度和周期。

如果产生溢出，则当两个捕捉的计数值相减后结果为负，并因此产生借位(将程序状态字(PSW)的第 0 位(CY)置 1)。如果发生这种情况，可以忽略 CY 并将计算结果作为脉冲宽度。此外，将 16 位定时器模式控制寄存器 00 (TMC00)的第 0 位(OVF00)清零(0)。

**图 6-49. 脉冲宽度测量时序示例 (2)**



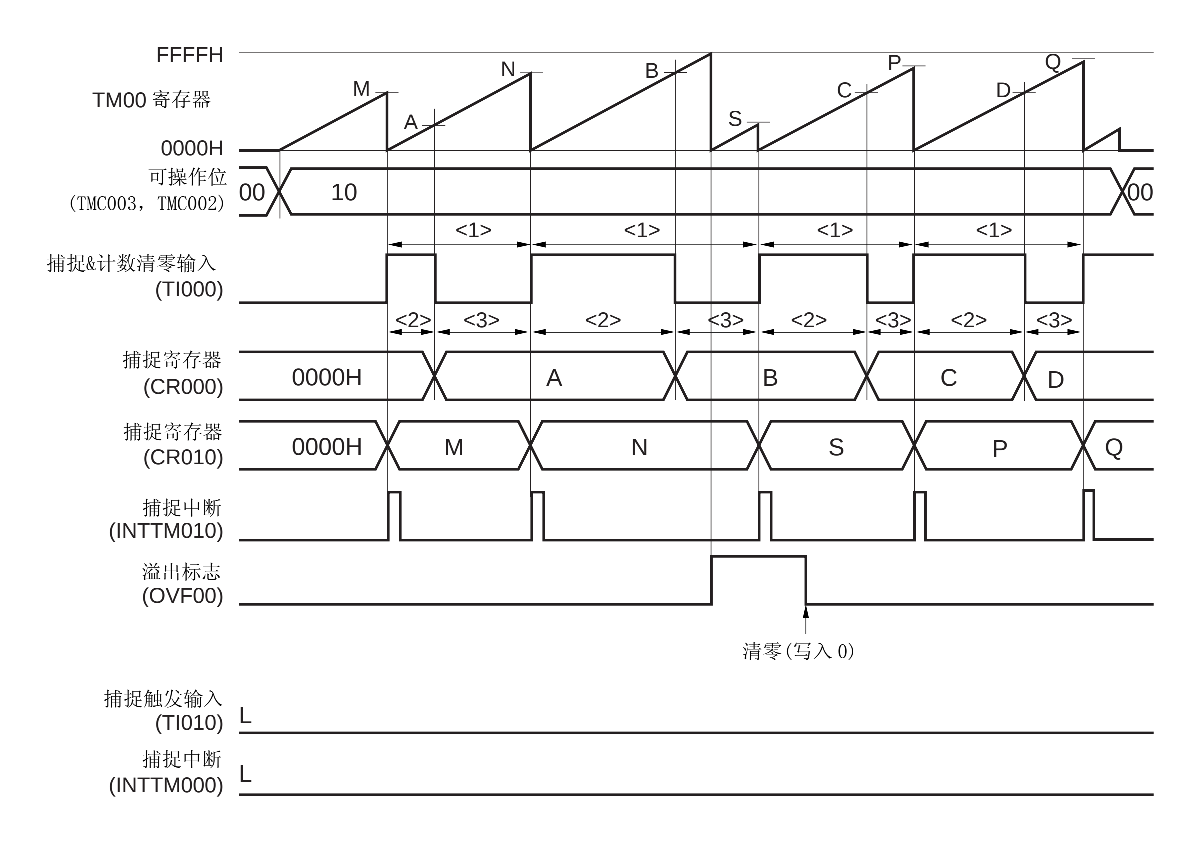
**(3)　通过使用 TI000 引脚的一个输入信号测量脉冲宽度(清零&启动模式(通过 TI000 引脚有效沿输入进入))**

设置清零&启动模式(通过 TI000 引脚有效沿进入)(TMC003 与 TMC002 = 10)。在 TI000 引脚检测到反向的有效沿时，将 TM00 的计数值捕捉到 CR000。当检测到 TI000 引脚的有效沿时，将 TM00 的计数值捕捉到 CR010，并将 TM00 清零(0000H)。因此，如果 TM00 没有溢出，则在 CR010 中存入一个周期。

如果产生溢出，则将 10000H 加上 CR010 的值，结果作为一个周期。此外，将 16 位定时器模式控制寄存器 00 (TMC00)的第 0 位(OVF00)清零(0)。

**图 6-50. 脉冲宽度测量时序示例 (3)**

• **TMC00 = 08H, PRM00 = 10H, CRC00 = 07H**



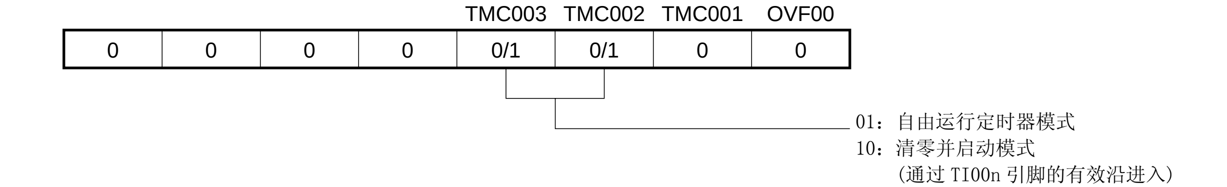

<1>　脉冲周期 =　(10000H × OVF00 位被设置为 1 的次数 +CR010 的捕捉值) × 计数时钟周期

<2>　高电平脉冲宽度 =　(10000H × OVF00 位被设置为 1 的次数+ CR000 的捕捉值) ×计数时钟周期

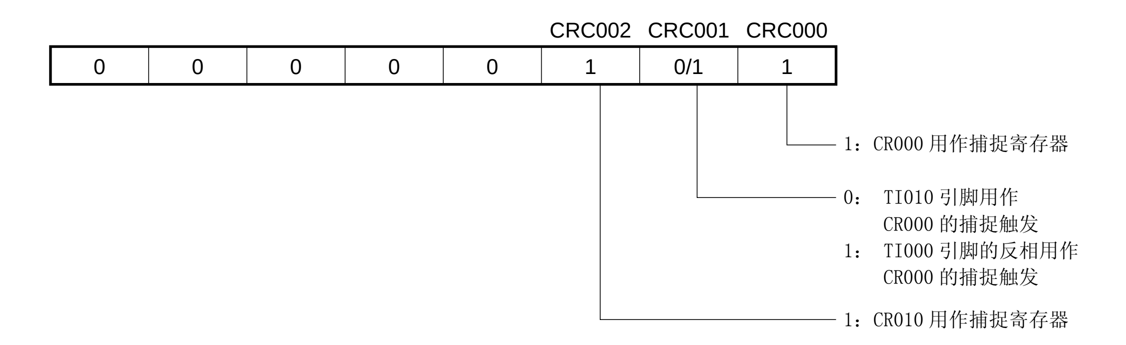
<3>　低电平脉冲宽度 =　(脉冲周期 – 高电平脉冲宽度)

**图 6-51. 脉冲宽度测量的寄存器设置示例 (1/2)**

**(a) 16 位定时器模式控制寄存器 00 (TMC00)**




**(b) 捕捉/比较控制寄存器 00 (CRC00)**




**(c) 16 位定时器输出控制寄存器 00 (TOC00)**

| | OSPT00 | OSPE00 | TOC004 | LVS00 | LVR00 | TOC001 | TOE00 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**(d) 预分频器模式寄存器 00 (PRM00)**

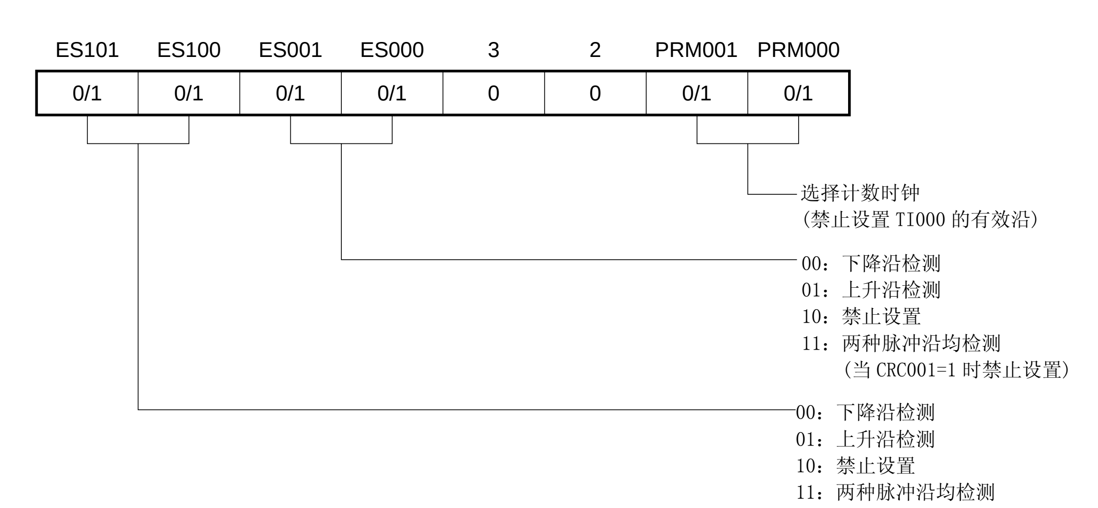


**图 6-51. 脉冲宽度测量的寄存器设置示例 (2/2)**

**(e) 16 位定时器计数器 00 (TM00)**

通过读取 TM00，可获得计数值。

**(f) 16 位捕捉/比较寄存器 000 (CR000)**

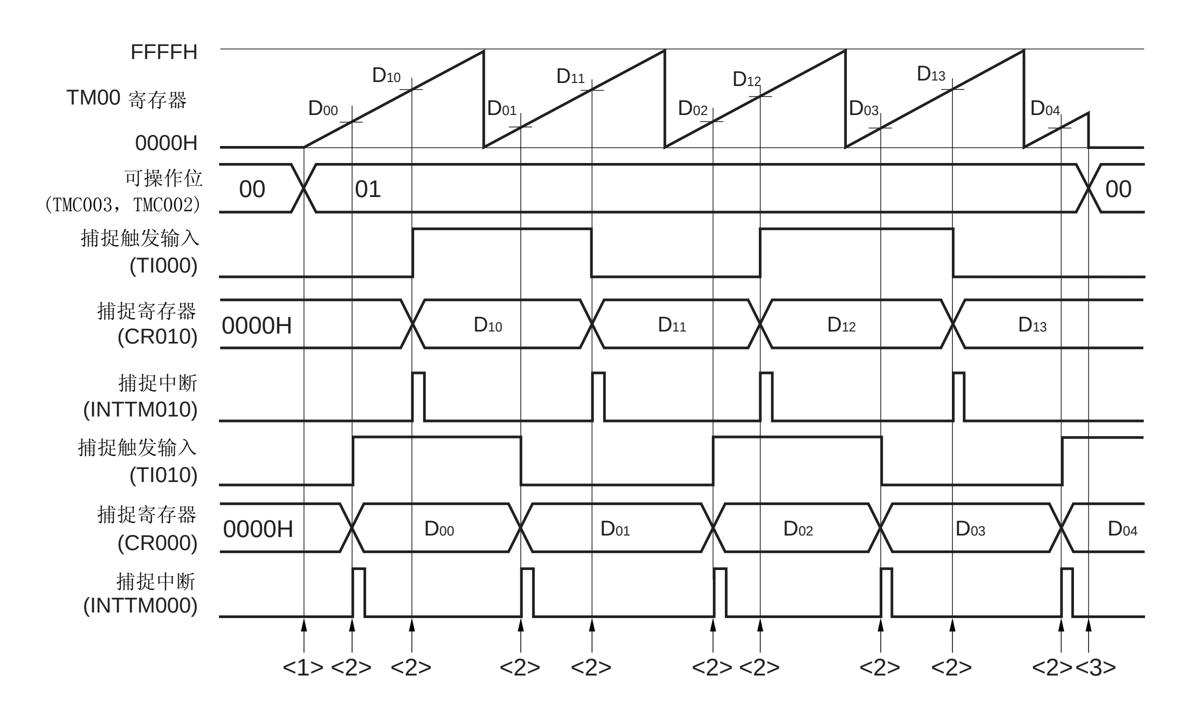
该寄存器用作捕捉寄存器。选择 TI000 或 TI010 引脚作为捕捉触发。当检测到指定脉冲沿的捕捉触发时，将 TM00 的计数值存入 CR000。

**(g) 16 位捕捉/比较寄存器 010 (CR010)**

该寄存器用作捕捉寄存器。将 TI000 引脚的输入信号用作捕捉触发。当检测到捕捉触发时，将 TM00 的计数值存入 CR010。

**图 6-52. 脉冲宽度测量软件处理示例 (1/2)**

**(a)** 自由运行定时器模式示例




**(b)** 清零&启动模式(通过 **TI000** 引脚有效沿进入)示例

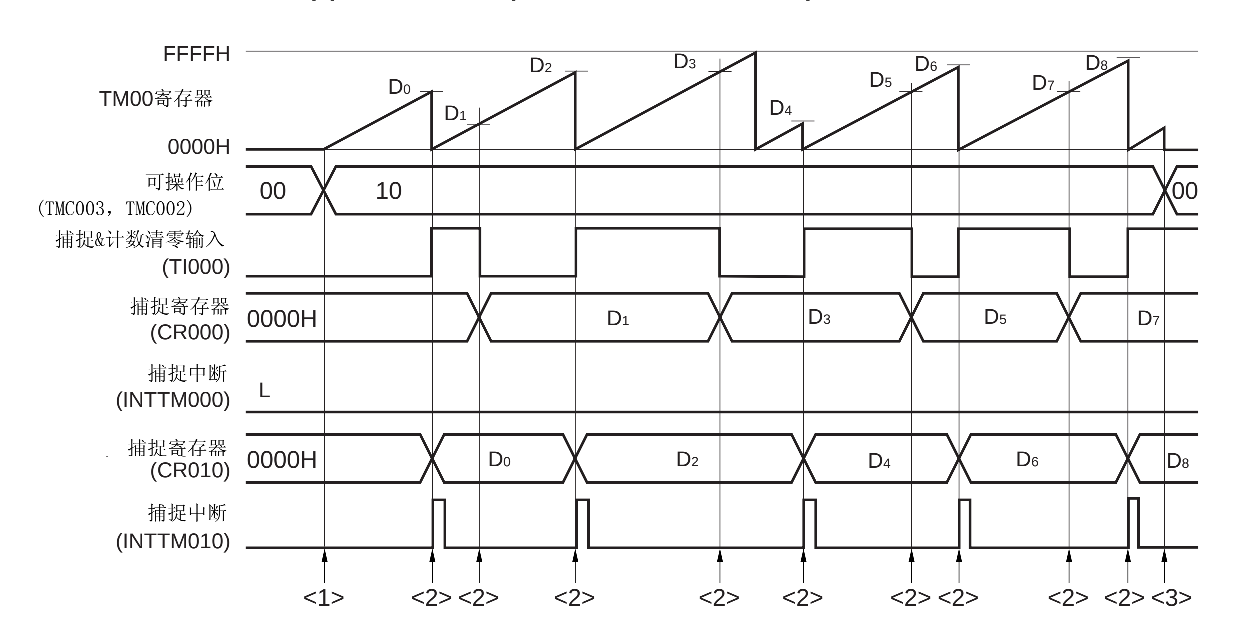
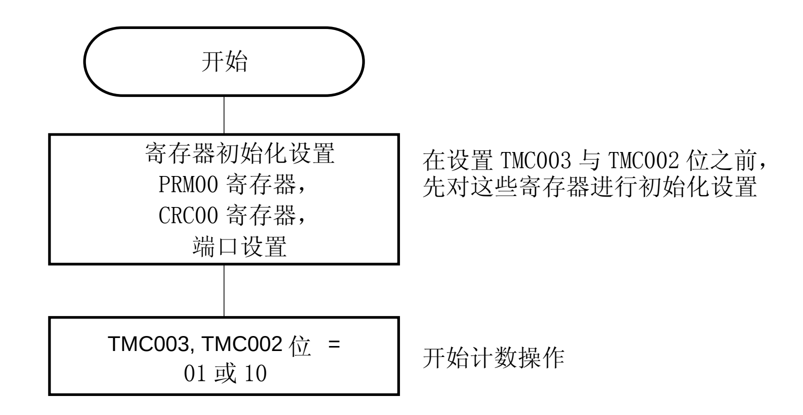
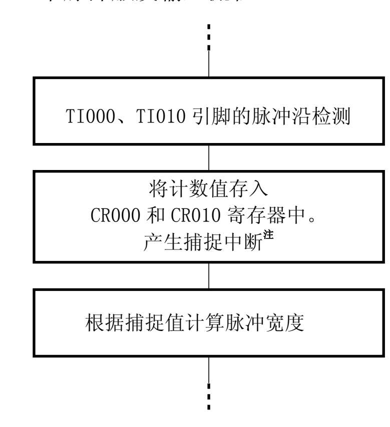


**图 6-52. 脉冲宽度测量软件处理示例 (2/2)**

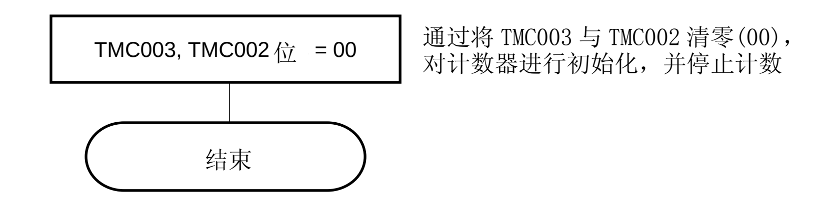
<1> 计数操作启动流程




<2> 单脉冲触发输入流程




<3> 计数操作停止流程




注 当选择 TI000 引脚输入的反向脉冲沿作为 CR000 的有效沿时，不产生捕捉中断信号(INTTM000)。

## 6.5 TM00 的特殊用途

### 6.5.1 TM00 操作期间重写 CR010

原则上，在 TM00 操作时(TMC003 与 TMC002 不等于 00)禁止重写 78K0/KC2 的 CR000 与 CR010(当它们用作比较寄存器时)的值。

但如果 CR010 用于 PPG 输出且占空比已被修改(在 CR010 的值与 TM00 的值相等后修改 CR010 的值。如果 CR010 的值与 TM0n 的值相等前立即修改 CR010，则可以会产生异常操作)，可以按下列过程修改 CR010，即使 TM00 正在操作。

修改 CR010 值的过程

<1>　禁止中断 INTTM010 (TMMK010 = 1).
<2>　当 TM00 的值与 CR010 的值相等时(TOC004 = 0)禁止反转定时器输出。
<3>　修改 CR010 的值
<4>　等待一个 TM00 计数时钟周期
<5>　允许 TM00 和 CR010（TOC004 = 1）相等时进行定时器输出反转操作。
<6>　将 INTTM010 的中断标志清零 (TMIF010 = 0)。
<7>　允许中断 INTTM010 (TMMK010 = 0).

**备注**　如需了解 TMIF010 和 TMMK010，可参见 **第十八章 中断功能**。

### 6.5.2 LVS00 和 LVR00 的设置

**(1) LVS00 和 LVR00 的使用**

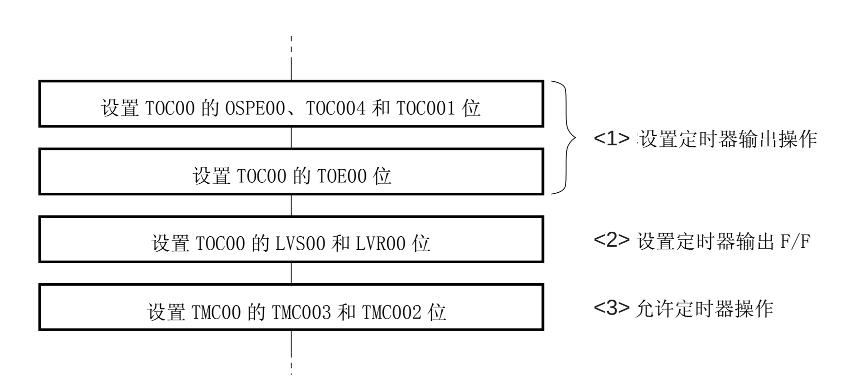
LVS00 与 LVR00 用于设置 TO00 引脚输出的默认值并可以在不需要定时器操作时(TMC003 与 TMC002 = 00)对定时器输出进行反转。当不需要软件控制时将 LVS00 与 LVR00 清零(00) (默认值: 低电平输出) 。

| LVS00 | LVR00 | 定时器输出状态 |
|---|---|---|
| 0 | 0 | 无变化 (低电平输出) |
| 0 | 1 | 清零 (低电平输出) |
| 1 | 0 | 设置 (高电平输出) |
| 1 | 1 | 禁止设置 |

**(2) LVS00 和 LVR00 的设置**

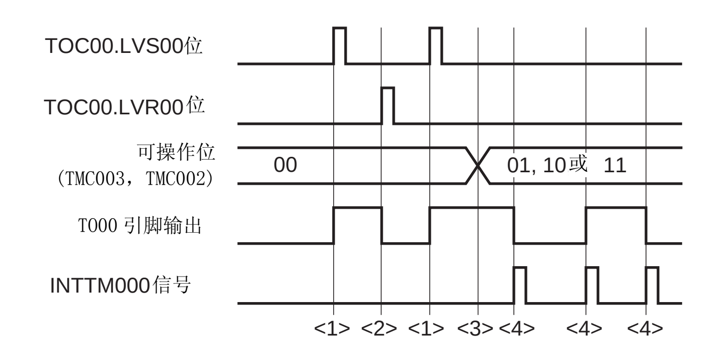
可按以下步骤设置 LVS00 和 LVR00。

**图 6-53. LVS0n 和 LVR0n 设置流程示例**




注意事项　**必须按上述<1>、<2>和<3>步骤设置 LVS00 和 LVR00。**
**步骤<2>可以在<1>之后<3>之前执行。**

**图 6-54. LVR0n 和 LVS0n 时序示例**




<1>　当 LVS00 与 LVR00 = 10 时将 TO00 引脚输出变为高电平。

<2>　当 LVS00 与 LVR00 = 01 时将 TO00 引脚输出变为低电平(如果 LVS00 与 LVR00 被清零(00)则引脚输出恒为高电平)。

<3>　当 TMC003 与 TMC002=01、10 或 11 时定时器开始操作。由于在开始操作前 LVS00 与 LVR00 被设置为 10，所以 TO00 引脚输出从高电平开始。定时器开始操作后，禁止对 LVS00 与 LVR00 进行设置，直至 TMC003 与 TMC002 = 00 (禁止定时器操作)。

<4>　每当产生中断信号(INTTM000)时，反转 TO00 引脚的输出电平。

## 6.6 16 位定时器/事件计数器 00 注意事项

**(1) 16 位定时器/事件计数器 00 每个通道的限制条件**

表 6-3 显示了每个通道的限制条件。

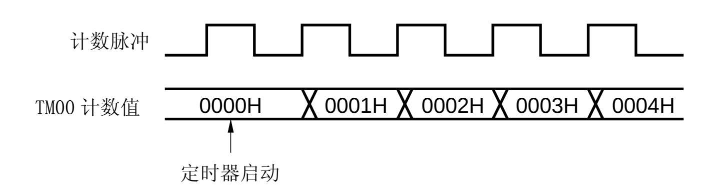
**表 6-3. 16 位定时器/事件计数器 00 每个通道的限制条件**

| 操作 | 限制条件 |
| --- | --- |
| 用作间隔定时器 | – |
| 用作方波输出 | |
| <R> 用作外部事件计数器 | |
| 用于清零&启动模式(通过 TI000 引脚有效沿输入进入) | 当使用检测 TI010 引脚有效沿时，禁止使用定时器输出(TO00)。<br>TOC00 = 00H |
| 用作由运行定时器 | – |
| <R> 用作 PPG 输出 | 0000H ≤ CR010 < CR000 ≤ FFFFH |
| <R> 用作单脉冲输出 | 禁止向 CR000 和 CR010 中设定相同的值 |
| 用于脉冲宽度测量 | 禁止使用定时器输出 (TO00) (TOC00 = 00H) |

**(2) 定时器启动误差**

定时器启动后，在产生相等信号之前可能发生将近一个时钟周期的误差。这是由于启动 TM00 计数操作与计数脉冲不同步引起的。

**图 6-55. TM00 计数启动时序**



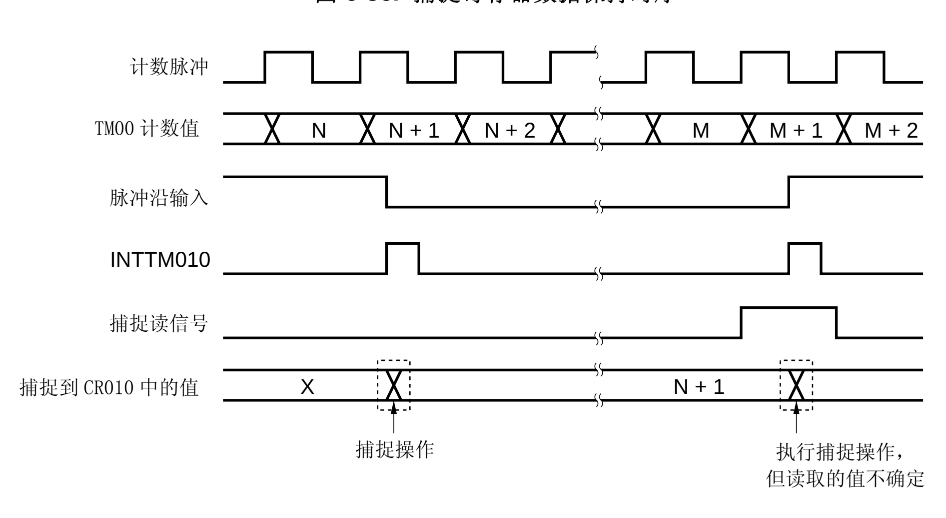

**(3) CR000 与 CR010 的设置 (清零&启动模式(TM00 与 CR000 相等时进入)**

为 CR000 与 CR010 设置 0000H 以外的值 (当 TM00 外部事件计数器时不能进行单脉冲计数)。

**(4) 捕捉寄存器数据保持时序**

(a) 如果在读取 CR000/CR010 时，有效沿输入到 TI000/TI010 引脚并且检测到 TI000 引脚脉冲的反向沿时，CR010 执行捕捉操作但 CR000/CR010 的读取值不确定。此时当检测到 TI000/TI010 引脚的有效沿时产生中断信号(INTTM000/INTTM010)(当检测到 TI000 引脚的反向脉冲沿时不产生中断信号)。
如果由于检测到 TI000/TI010 引脚的有效沿而捕捉到计数值时，则应在产生 INTTM000/INTTM010 信号后读取 CR000/CR010 的值。

**图 6-56. 捕捉寄存器数据保持时序**




(b) 当 16 位定时器/事件计数器 00 停止操作后，CR000 与 CR010 的值不确定。

**(5) 有效沿的设置**

当定时器停止操作时(TMC003 与 TMC002 = 00)设置 TI000 引脚的有效沿。使用 ES000 与 ES001 设置有效沿。

**(6) 单脉冲的重新触发**

必须确定在单脉冲输出模式下正在输出一个活动电平时，没有触发产生。而且必须在输出当前活动电平后输入下一个触发。

## (7) OVF00 标志的操作

### (a) OVF00 标志的设置 (1)

在以下情况下以及 TM00 溢出时将 OVF00 标志设置为 1。

选择清零&启动模式(在 TM00 与 CR000 相等时进入)。

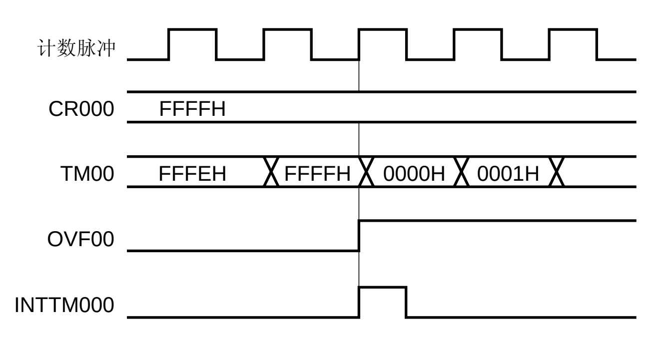
↓

设置 CR000 = FFFFH.

↓

当 TM00 与 CR000 相等且 TM00 从 FFFFH 变为 0000H 时 TM00 被清零

**图 6-57. OVF00 标志操作时序**




### (b) OVF00 标志清零

即使在 TM00 溢出后下一个计数时钟被计数之前(TM00 的值变为 0001H 之前)将 OVF00 标志清零(0)，OVF00 也会再次被设置为 1，并且对其清零是无效的。

## (8) 单脉冲输出

在自由运行定时器模式下或者在清零&启动模式(通过 TI000 引脚有效沿进入)下，单脉冲输出能正确操作。而在清零&启动模式(在 TM00 与 CR000 相等时进入)下不能输出单脉冲。

**(9)** 捕捉操作

**(a)** 当指定 **TI000** 的有效沿作为计数时钟时

当指定 TI000 的有效沿作为计数时钟时，捕捉寄存器(指定 TI000 作为触发)不能正确操作。

**(b)** 通过 **TI010** 与 **TI000** 引脚输入信号准确捕捉计数值的脉冲宽度

为了准确捕捉计数值，作为捕捉触发的 TI000 与 TI010 引脚的输入脉冲宽度必须大于两个计数时钟(由 PRM00 选择)(参见图 **6-7**)。

**(c)** 中断信号的产生

在计数时钟的下降沿进行捕捉操作，但中断信号(INTTM000 与 INTTM010)则在下一个计数时钟的上升沿产生。(参见图 **6-7**)。

**(d)** 当 **CRC001 (**捕捉**/**比较控制寄存器 **00 (CRC00)**的第 **1** 位**)=1** 时的注意事项

在 TI000 引脚输入信号的反向沿将 TM00 的计数值捕捉到 CR000 时，捕捉到计数值后不产生中断信号(INTTM000)。如果在此操作期间，从 TI010 引脚检测到有效沿，则不执行捕捉操作但会产生中断信号(INTTM000)作为外部中断信号。当不使用外部中断时屏蔽 INTTM000 信号。

**(10)** 脉冲沿检测

**(a)** 复位后指定有效沿

如果复位后 TI000 或 TI010 引脚处于高电平，且当指定上升沿或兼有两种脉冲沿作为 TI000 或 TI010 引脚的有效沿时，允许 16 位定时器/事件计数器 00 的操作，则检测到 TI000 或 TI010 引脚的高电平作为上升沿。上拉 TI000 或 TI010 引脚时要特别注意。但如果操作停止之后又允许操作，则不检测上升沿。

**(b)** 采样时钟**(**用于消除噪音**)**

根据 TI000 的有效沿是用作计数时钟还是用作捕捉触发，采样时钟(用于消除噪音)会发生变化。在前一种情况中采样时钟恒为 $f_{PRS}$。而在后一种情况中则将由 PRM00 选择的计数时钟用于采样。

当采样到 TI000 引脚的输入信号并且在一个周期内两次检测到有效电平时，才认为检测到有效沿。因此可以消除窄脉冲宽度的噪音(参见图 **6-7**)。

**(11)** 定时器操作

无论 CPU 处于何种操作模式，当定时器停止操作时，不响应 TI000/TI010 引脚的输入信号。

备注　$f_{PRS}$: 外围硬件时钟频率

# 第七章　8 位定时器/事件计数器 50 和 51

## 7.1 8 位定时器/事件计数器 50 和 51 的功能

8 位定时器/事件计数器 50 和 51 具有以下功能。

- 间隔定时器
- 外部事件计数器
- 方波输出
- PWM 输出

## 7.2 8 位定时器/事件计数器 50 和 51 的构成

8 位定时器/事件计数器 50 和 51 包括以下硬件。

**表 7-1. 8 位定时器/事件计数器 50 和 51 的构成**

| 项目 | 配置 |
| --- | --- |
| 定时器寄存器 | 8 位定时器计数器 5n (TM5n) |
| 寄存器 | 8 位定时器比较寄存器 5n (CR5n) |
| 定时器输入 | TI5n |
| 定时器输出 | TO5n |
| 控制寄存器 | 定时器时钟选择寄存器 5n (TCL5n)<br>8 位定时器模式控制寄存器 5n (TMC5n)<br>端口模式寄存器 1 (PM1)或端口模式寄存器 3 (PM3)<br>端口寄存器 1 (P1)或端口寄存器 3 (P3) |

图 7-1 与图 7-2 显示 8 位定时器/事件计数器 50 和 51 的框图

**图 7-1. 8 位定时器/事件计数器 50 的框图**

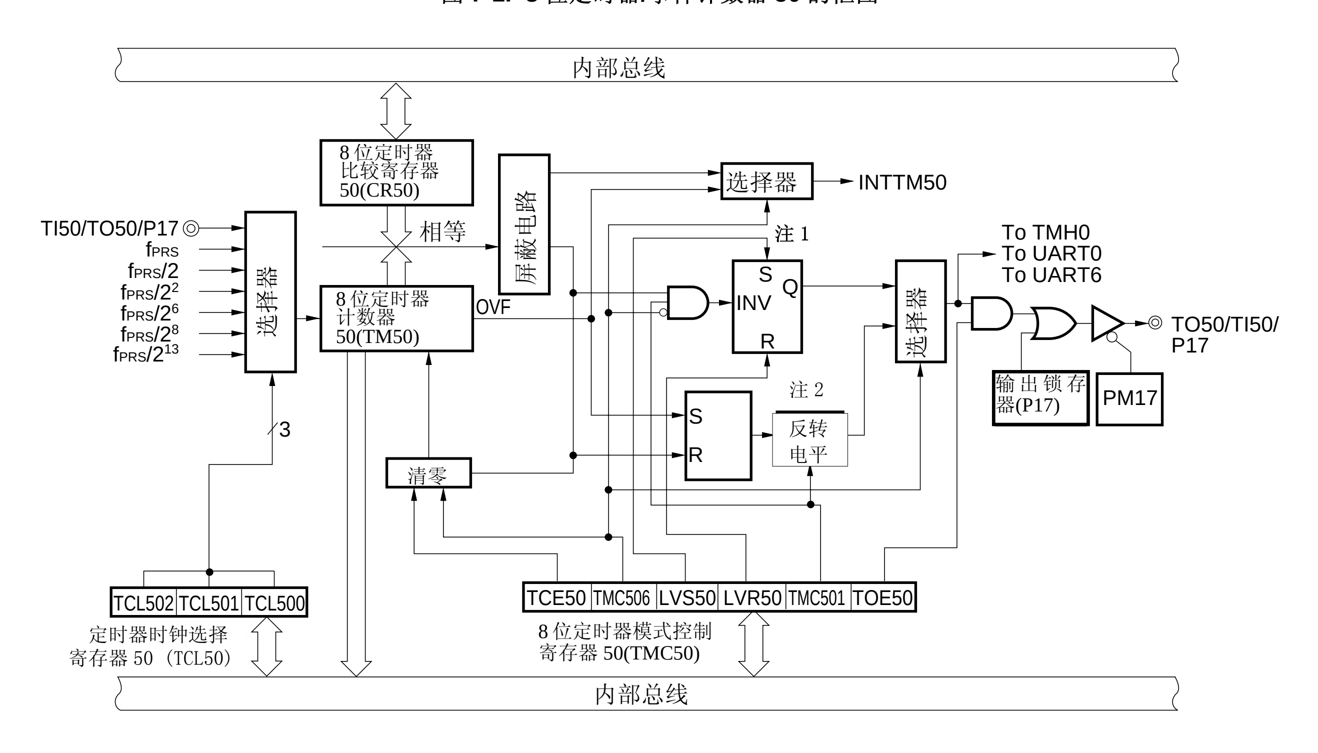
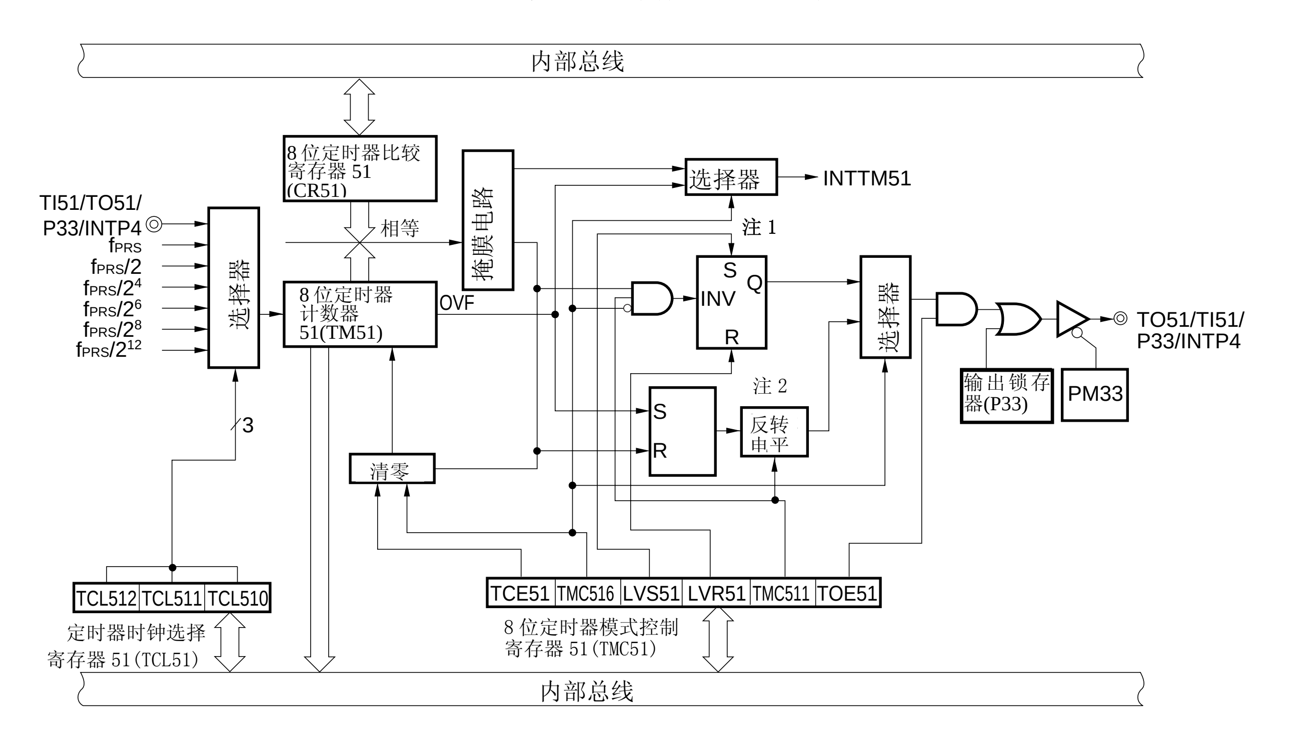



**图 7-2. 8 位定时器/事件计数器 51 的框图**




注 1. 定时器输出 F/F
 2. PWM 输出 F/F

**(1) 8 位定时器计数器 5n (TM5n)**

TM5n 是一个 8 位寄存器，用于对计数脉冲进行计数，它是只读的。

该计数器的计数值随计数时钟的上升沿同步增加。

**图 7-3. 8 位定时器计数器 5n (TM5n)的格式**

地址：FF16H (TM50), FF1FH (TM51)　　复位后：00H　　R

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TM5n<br>(n = 0, 1) | | | | | | | | |

在以下几种情况下将计数值清零(00H)。

<1> 产生复位信号

<2> TCE5n 清零

<3> 在 TM5n 与 CR5n 相等时清零和启动模式中，当 TM5n 与 CR5n 相等时。

**(2) 8 位定时器比较寄存器 5n (CR5n)**

可以由 8 位存储器操作指令对 CR5n 进行读写。

除了 PWM 模式外，CR5n 的值将与 8 位定时器计数器 5n(TM5n)的计数值频繁相比较，如果相等将产生中断请求(INTTM5n)。

在 PWM 模式下，TM5n 与 CR5n 的值相等时，TO5n 输出非有效电平，但不产生中断信号。

CR5n 的取值范围为：00H ～ FFH。

复位信号的产生将 CR5n 清零(00H)。

**图 7-4. 8 位定时器比较寄存器 5n (CR5n)的格式**

地址：FF17H (CR50), FF41H (CR51)　　复位后：00H　　R/W



| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CR5n<br>(n = 0, 1) | | | | | | | | |

注意事项 **1. 在 TM5n 与 CR5n (TMC5n6 = 0)相等时清零和启动模式下，不要将其它值写入 CR5n。**

**2. 在 PWM 模式中，以至少 3 个计数时钟(由 TCL5n 选择时钟)的间隔对 CR5n 进行重写。**

备注　n = 0, 1

## 7.3 控制 8 位定时器/事件计数器 50 和 51 的寄存器

以下四种寄存器用于控制 8 位定时器/事件计数器 50 和 51。

- 定时器时钟选择寄存器 5n (TCL5n)
- 8 位定时器模式控制寄存器 5n (TMC5n)
- 端口模式寄存器 1 (PM1)或端口模式寄存器 3 (PM3)
- 端口寄存器 1 (P1)或端口寄存器 3 (P3)

### (1) 定时器时钟选择寄存器 5n (TCL5n)

该寄存器设置 8 位定时器/事件计数器 5n 的计数时钟，和 TI5n 引脚输入的有效沿。
可由 1 位或 8 位存储器操作指令设置 TCL5n。
复位信号的产生将 TCL5n 清零(00H)。

备注　n = 0, 1

**图 7-5. 定时器时钟选择寄存器 50 (TCL50) 的格式**

地址: FF6AH　复位后: 00H　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TCL50 | 0 | 0 | 0 | 0 | 0 | TCL502 | TCL501 | TCL500 |

| TCL502 | TCL501 | TCL500 | 计数时钟选择 | | | | |
|---|---|---|---|---|---|---|---|
| | | | | $f_{PRS}$ = 2 MHz | $f_{PRS}$ = 5 MHz | $f_{PRS}$ = 10 MHz | $f_{PRS}$ = 20 MHz |
| 0 | 0 | 0 | TI50 引脚下降沿 | | | | |
| 0 | 0 | 1 | TI50 引脚上升沿 | | | | |
| 0 | 1 | 0 | $f_{PRS}$ | 2 MHz | 5 MHz | 10 MHz | 20 MHz |
| 0 | 1 | 1 | $f_{PRS}/2$ | 1 MHz | 2.5 MHz | 5 MHz | 10 MHz |
| 1 | 0 | 0 | $f_{PRS}/2^2$ | 500 kHz | 1.25 MHz | 2.5 MHz | 5 MHz |
| 1 | 0 | 1 | $f_{PRS}/2^6$ | 31.25 kHz | 78.13 kHz | 156.25 kHz | 312.5 kHz |
| 1 | 1 | 0 | $f_{PRS}/2^8$ | 7.81 kHz | 19.53 kHz | 39.06 kHz | 78.13 kHz |
| 1 | 1 | 1 | $f_{PRS}/2^{13}$ | 0.24 kHz | 0.61 kHz | 1.22 kHz | 2.44 kHz |

**注意事项 1. 如果要向 TCL50 写入其它值，则必须提前停止定时器操作。**
**2. 必须将第 3 ～ 7 位清零。**

备注　$f_{PRS}$: 外部设备时钟频率

**图 7-6. 定时器时钟选择寄存器 51 (TCL51) 的格式**

地址: FF8CH 复位后: 00H R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TCL51 | 0 | 0 | 0 | 0 | 0 | TCL512 | TCL511 | TCL510 |

| TCL512 | TCL511 | TCL510 | 计数时钟选择 | | | | |
|---|---|---|---|---|---|---|---|
| | | | | $f_{PRS}$ = 2 MHz | $f_{PRS}$ = 5 MHz | $f_{PRS}$ = 10 MHz | $f_{PRS}$ = 20 MHz |
| 0 | 0 | 0 | TI51 下降沿 | | | | |
| 0 | 0 | 1 | TI51 上升沿 | | | | |
| 0 | 1 | 0 | $f_{PRS}$ | 2 MHz | 5 MHz | 10 MHz | 20 MHz |
| 0 | 1 | 1 | $f_{PRS}/2$ | 1 MHz | 2.5 MHz | 5 MHz | 10 MHz |
| 1 | 0 | 0 | $f_{PRS}/2^4$ | 125 kHz | 312.5 kHz | 625 kHz | 1.25 MHz |
| 1 | 0 | 1 | $f_{PRS}/2^6$ | 31.25 kHz | 78.13 kHz | 156.25 kHz | 312.5 kHz |
| 1 | 1 | 0 | $f_{PRS}/2^8$ | 7.81 kHz | 19.53 kHz | 39.06 kHz | 78.13 kHz |
| 1 | 1 | 1 | $f_{PRS}/2^{12}$ | 0.49 kHz | 1.22 kHz | 2.44 kHz | 4.88 kHz |

**注意事项** **1.** 如果要向 **TCL51** 写入其它值，则必须提前停止定时器操作。

**2.** 必须将第 **3** ～ **7** 位清零。

**备注** $f_{PRS}$: 外部设备时钟频率

**(2) 8 位定时器模式控制寄存器 5n (TMC5n)**

TMC5n 主要执行以下五种设置操作。

<1>　8 位定时器计数器 5n (TM5n)计数操作控制
<2>　8 位定时器计数器 5n (TM5n)操作模式选择
<3>　定时器输出 F/F (flip flop 触发器)状态设置
<4>　定时器 F/F 控制或 PWM(自由运行)模式中有效电平的选择
<5>　定时器输出控制

可以由 1 位或 8 位存储器操作指令设置 TMC5n。
复位信号的产生将该寄存器清零(00H)。

备注　n = 0, 1

**图 7-7. 8 位定时器模式控制寄存器 50 (TMC50)的格式**

地址: FF6BH　复位后: 00H　R/W[注]

| 符号 | <7> | 6 | 5 | 4 | <3> | <2> | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| TMC50 | TCE50 | TMC506 | 0 | 0 | LVS50 | LVR50 | TMC501 | TOE50 |

| TCE50 | TM50 计数操作控制 |
|---|---|
| 0 | 清零后，禁止计数操作 (计数器停止操作) |
| 1 | 开始计数 |

| TMC506 | TM50 操作模式选择 |
|---|---|
| 0 | TM50 与 CR50 相等时清零和启动模式 |
| 1 | PWM (自由运行)模式 |

| LVS50 | LVR50 | 定时器输出 F/F 状态设置 |
|---|---|---|
| 0 | 0 | 无变化 |
| 0 | 1 | 定时器输出 F/F 复位(0) (TO50 默认输出：低电平) |
| 1 | 0 | 定时器输出 F/F 设置(1) (TO50 默认输出：高电平) |
| 1 | 1 | 禁止设置 |

| TMC501 | 其他模式(TMC506 = 0) | PWM 模式(TMC506 = 1) |
|---|---|---|
| | 定时器 F/F 控制 | 有效电平选择 |
| 0 | 禁止反转操作 | 高电平有效 |
| 1 | 允许反转操作 | 低电平有效 |

| TOE50 | 定时器输出控制 |
|---|---|
| 0 | 禁止输出 (TM50 输出为低电平) |
| 1 | 允许输出 |

注　第 2 位和第 3 位为只写操作位。

(参见下页的**注意事项**和**备注**)

**图 7-8. 8 位定时器模式控制寄存器 51 (TMC51) 的格式**

地址: FF43H 复位后: 00H R/W[注]

| 符号 | <7> | 6 | 5 | 4 | <3> | <2> | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| TMC51 | TCE51 | TMC516 | 0 | 0 | LVS51 | LVR51 | TMC511 | TOE51 |

| TCE51 | TM51 计数操作控制 |
|---|---|
| 0 | 清零后禁止计数操作 (计数器停止操作) |
| 1 | 开始计数 |

| TMC516 | TM51 操作模式选择 |
|---|---|
| 0 | TM51 与 CR51 相等时清零和启动模式 |
| 1 | PWM (自由运行)模式 |

| LVS51 | LVR51 | 定时器输出 F/F 状态设置 |
|---|---|---|
| 0 | 0 | 无变化 |
| 0 | 1 | 定时器输出 F/F 复位 (0) (TO51 默认输出：低电平) |
| 1 | 0 | 定时器输出 F/F 设置 (1) (TO51 默认输出：高电平) |
| 1 | 1 | 禁止设置 |

| TMC511 | 其他模式 (TMC516 = 0) | PWM 模式 (TMC516 = 1) |
|---|---|---|
| | 定时器 F/F 控制 | 有效电平选择 |
| 0 | 禁止反转操作 | 高电平有效 |
| 1 | 允许反转操作 | 低电平有效 |

| TOE51 | 定时器输出控制 |
|---|---|
| 0 | 禁止输出 (TM51 输出为低电平) |
| 1 | 允许输出 |

注 第 2 位和第 3 位为只写操作位。

**注意事项 1.除 PWM 模式外，对 LVS5n 与 LVR5n 的设置有效。**

**2.按下列顺序执行<1> ～ <4>，注意不能同时执行。**

**<1>设置 TMC5n1，TMC5n6: 操作模式设置**

**<2>设置 TOE5n，允许输出: 允许定时器输出**

**<3>设置 LVS5n, LVR5n (见注意事项 1): 定时器 F/F 设置**

**<4>设置 TCE5n**

<R> **3.当 TCE5n = 1 时，禁止设定 TMC5n 的其它位。**

备注 1. 在 PWM 模式中，将 TCE5n 清零可禁止 PWM 输出。

2. 读取 LVS5n 与 LVR5n 时，读取的值为 0。
3. 无论 TCE5n 取何值，TMC5n6、LVS5n、 LVR5n、 TMC5n1 和 TOE5n 的值都与 TO5n 引脚有关。
4. n = 0, 1

**(3) 端口模式寄存器 1 和 3 (PM1, PM3)**

这两个寄存器用于按位设置端口 1 和 3 的输入输出操作模式。

当 P17/TO50/TI50 和 P33/TO51/TI51/INTP4 引脚用于定时器输出时，将 PM17 和 PM33 以及 P17 与 P33 的输出锁存器清零。

当 P17/TO50/TI50 和 P33/TO51/TI51/INTP4 引脚用于定时器输入时，将 PM17 和 PM33 置 1，P17 与 P33 的输出锁存值可以为 0 也可以为 1。

可以由 1 位或 8 位存储器操作指令设置 PM1 和 PM3。

复位信号的产生将寄存器的内容设置为 FFH。

**图 7-9. 端口模式寄存器 1 (PM1) 的格式**

地址: FF21H  复位后: FFH  R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM1 | PM17 | PM16 | PM15 | PM14 | PM13 | PM12 | PM11 | PM10 |

| PM1n | P1n 引脚 I/O 模式选择 (n = 0 ～ 7) |
|---|---|
| 0 | 输出模式 (输出缓冲器打开) |
| 1 | 输入模式 (输出缓冲器关闭) |

**图 7-10. 端口模式寄存器 3 (PM3)的格式**

地址: FF23H  复位后: FFH  R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM3 | 1 | 1 | 1 | 1 | PM33 | PM32 | PM31 | PM30 |

| PM3n | P3n 引脚 I/O 模式选择(n = 0 ～ 3) |
|---|---|
| 0 | 输出模式 (输出缓冲器打开) |
| 1 | 输入模式 (输出缓冲器关闭) |

## 7.4 8 位定时器/事件计数器 50 和 51 的操作

### 7.4.1 用作间隔定时器

8 位定时器/事件计数器 5n 用作间隔定时器，可以在间隔时间内不断产生中断请求，该间隔时间由 8 位定时器比较寄存器 5n(CR5n)的预置值决定。

当 8 位定时器计数器 5n(TM5n)中的计数值与 CR5n 中的值相相等时，将 TM5n 清零并继续计数，同时产生中断请求信号(INTTM5n)。

根据定时器时钟选择寄存器 5n(TCL5n)的第 0～2 位(TCL5n0 ～ TCL5n2)选择 TM5n 的计数时钟。

设置

<1> 设置寄存器

- TCL5n: 设置计数时钟
- CR5n: 比较值
- TMC5n: 停止计数操作，选择 TM5n 与 CR5n 相等时清零和启动模式。
  (TMC5n = 0000×××0B， × = 不必考虑)

<2> 将 TCE5n 置 1 后，开始计数。

<3> 如果 TM5n 与 CR5n 值相等时，产生中断 INTTM5n (将 TM5n 清零(00H))。

<4> 以相同的时间间隔重复产生 INTTM5n。

将 TCE5n 清零，停止计数操作。

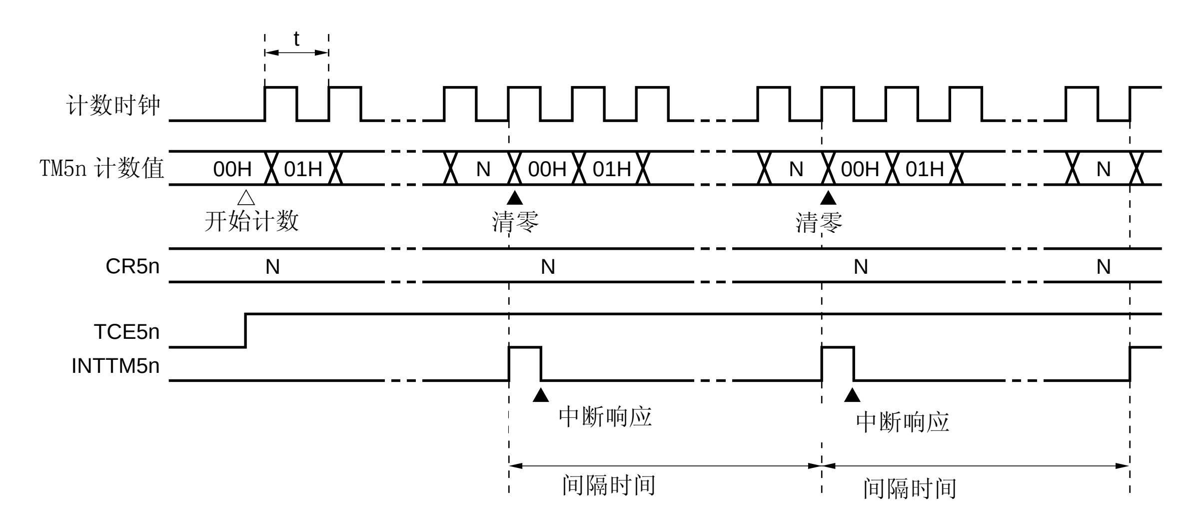
**注意事项 操作期间不要向 CR5n 写入其它值。**

**备注** **1.** 如需了解如何允许 INTTM5n 中断，请参见 **第十八章 中断功能**。

**2.** n = 0, 1

**图 7-11. 间隔定时器操作的时序 (1/2)**

**(a) 基本操作**



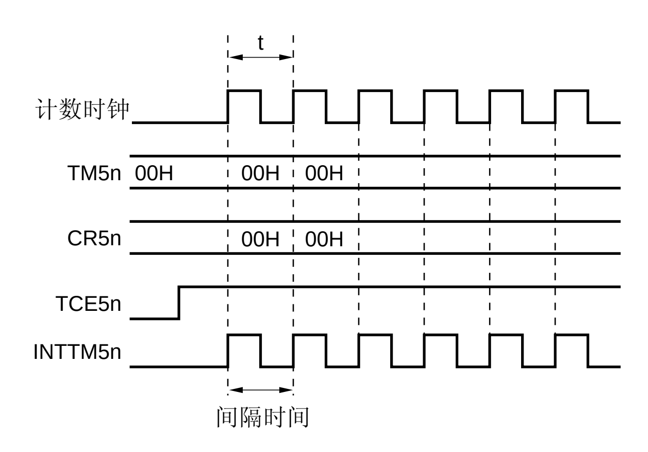

**备注** 间隔时间 = (N + 1) × t

N = 01H ～ FEH

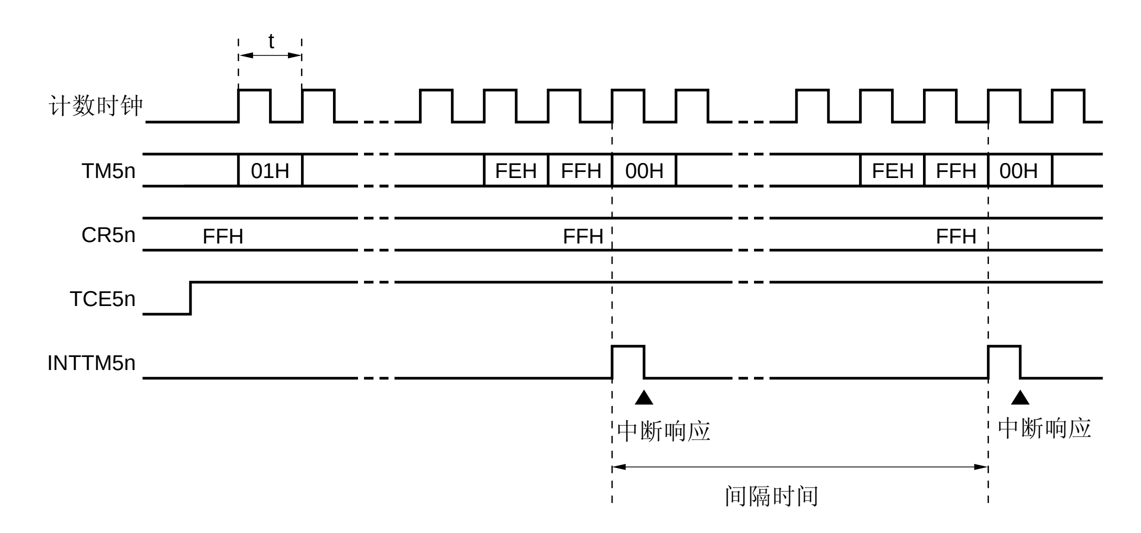
n = 0, 1

图 7-11. 间隔定时器操作的时序(2/2)

(b) CR5n = 00H




(c) CR5n = FFH




备注 n = 0, 1

### 7.4.2 用作外部事件计数器

外部事件计数器通过 8 位定时器计数器 5n（TM5n）对输入到 TI5n 引脚的外部时钟脉冲计数。

TM5n 随输入的有效沿次数递增。有效沿由定时器时钟选择寄存器 5n（TCL5n）规定，可以选择上升沿或下降沿。

当 TM5n 中的计数值与 8 位定时器比较寄存器 5n(CR5n)中的值相相等时，将 TM5n 清零，并产生中断请求信号(INTTM5n)。

一旦 TM5n 中的值与 CR5n 的值相等, 将产生 INTTM5n 中断请求。

设置

<1> 设置寄存器

- 设置端口模式寄存器 (PM17 或 PM33)[注] 的值为 1。
- TCL5n: 选择 TI5n 引脚输入脉冲沿
  TI5n 引脚下降沿 → TCL5n = 00H
  TI5n 引脚上升沿 → TCL5n = 01H
- CR5n: 比较值
- TMC5n: 停止计数，选择 TM5n 与 CR5n 相等时清零和启动模式，禁止定时器 F/F 反转操作，禁止定时器输出。

<R> (TMC5n = 00000000B)

<2> 当 TCE5n = 1 时，开始对从 TI5n 引脚输入的脉冲计数。

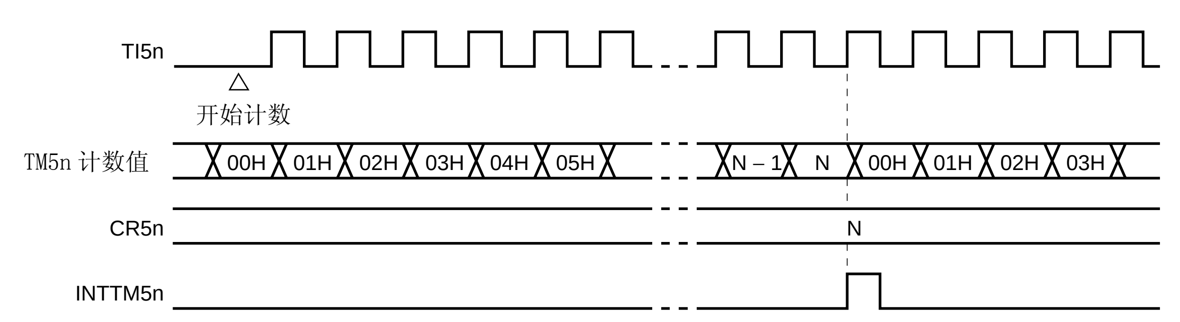
<3> 当 TM5n 与 CR5n 的值相等时，将产生 INTTM5n (TM5n 清零(00H))。

<4> 在经过以上设置后，每当 TM5n 与 CR5n 的值相等时都会产生 INTTM5n 中断请求。

**注** 8 位定时器/事件计数器 50: PM17
8 位定时器/事件计数器 51: PM33

**备注** 如需了解如何允许 INTTM5n 中断，请参见 **第十八章 中断功能**。

**图 7-12. 外部事件计数器操作的时序 (规定上升沿有效)**




**备注** N = 00H ～ FFH
n = 0, 1

### 7.4.3 方波输出操作

以 8 位定时器比较寄存器 5n（CR5n）的预置值决定的间隔时间输出任选频率的方波。

通过将 8 位定时器模式控制寄存器 5n(TMC5n)的第 0 位(TOE5n)置 1，可以在规定的间隔时间内反转 TO5n 引脚输出的状态，该间隔时间由 8 位定时器比较寄存器 5n(CR5n)的预置值决定。这样，可以输出任选频率的方波(占空比＝50%)。

设置

<1> 设置寄存器

- 端口输出锁存器 (P17 或 P33)[注] 和端口模式寄存器 (PM17 或 PM33)[注] 清零。
- TCL5n: 选择计数时钟
- CR5n: 比较值
- TMC5n: 停止计数，选择 TM5n 与 CR5n 相等时清零和启动模式

| LVS5n | LVR5n | 定时器输出 F/F 状态设置 |
|---|---|---|
| 1 | 0 | 定时器输出 F/F 清零(0) (TO50 引脚的默认输出值：低电平) |
| 0 | 1 | 定时器输出 F/F 置 (1) (TO5n 引脚的默认输出值：高电平) |

允许定时器输出
(TMC5n = 00001011B 或 00000111B)

<2> 一旦 TCE5n = 1，便开始计数。

<3> 通过 TM5n 与 CR5n 的相等，反转定时器输出 F/F。产生 INTTM5n 后，将 TM5n 清零(00H)。

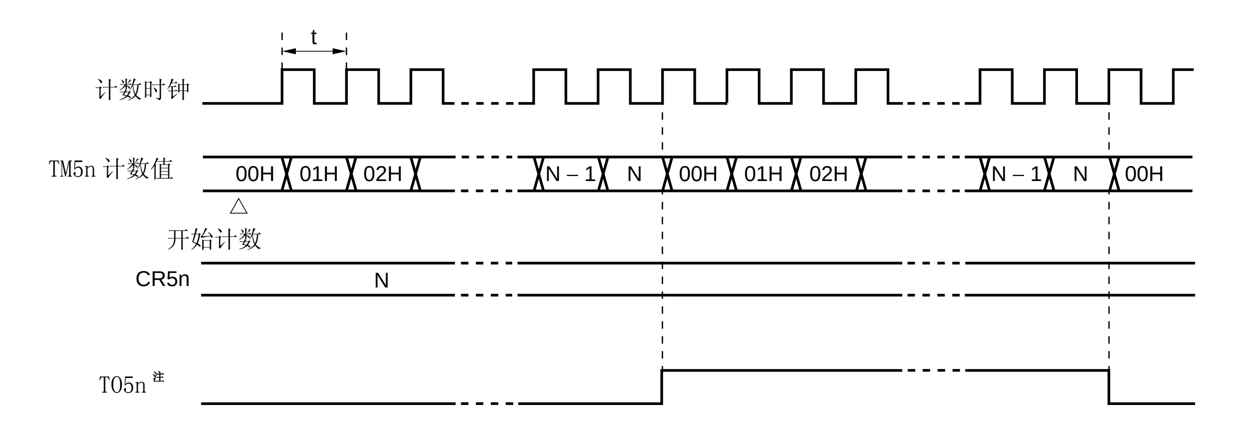
<4> 在经过以上设置后，在相同间隔内反转定时器输出 F/F，并从 TO5n 输出方波。
频率如下。
频率 = 1/2t (N + 1)
(N: 00H ～ FFH)

注 8 位定时器/事件计数器 50: P17, PM17
8 位定时器/事件计数器 51: P33, PM33

**注意事项 操作期间不要向 CR5n 写入其它值。**

备注 **1.** 如需了解如何允许 INTTM5n 中断，请参见 **第十八章 中断功能。**
**2.** n = 0, 1

图 7-13. 方波输出操作的时序




**注** 由 8 位定时器模式控制寄存器 5n(TMC5n)的第 2、3 位(LVR5n, LVS5n)设置 TO5n 的初始值。

### 7.4.4 PWM 输出操作

当 8 位定时器模式控制寄存器 5n(TMC5n)的第 6 位(TMC5n6)置 1 时，8 位定时器/事件计数器 5n 作为 PWM 输出使用。

占空比由 8 位定时器比较寄存器 5n(CR5n)的值决定，并从 TO5n 引脚输出。

设置 CR5n 为 PWM 脉冲的有效电平宽度；可以由 TMC5n 的第 1 位(TMC5n1)选择有效电平。

由定时器时钟选择寄存器 5n(TCL5n)的第 0～2 位(TCL5n0 ～ TCL5n2)选择计数时钟。

可以由 TMC5n 的第 0 位(TOE5n)决定 PWM 输出允许/禁止。

**注意事项 在 PWM 模式中，当对 CR5n 进行重写时，需要至少 3 个计数时钟(该计数时钟由 TCL5n 选择)周期的写间隔。**

**备注** n = 0, 1

### (1) PWM 输出的基本操作

设置

<1> 设置寄存器

- 将端口输出锁存器 (P17 或 P33)[注] 和端口模式寄存器 (PM17 或 PM33)[注] 清零。
- TCL5n: 选择计数时钟
- CR5n: 比较值
- TMC5n: 停止计数操作，选择 PWM 模式

  定时器输出 F/F 不改变。

| TMC5n1 | 有效电平选择 |
|---|---|
| 0 | 高电平有效 |
| 1 | 低电平有效 |

允许定时器输出

(TMC5n = 01000001B 或 01000011B)

<2> 当 TCE5n = 1 时，计数开始。

TCE5n 清零可以停止计数操作。

注 8 位定时器/事件计数器 50: P17, PM17

8 位定时器/事件计数器 51: P33, PM33

PWM输出操作

<1> PWM 输出(从 TO5n 输出)一直输出非有效电平，直至产生溢出。

<2> 当发生溢出时，输出有效电平。直至 CR5n 与 8 位定时器计数器 5n（TM5n）的计数值相等时才停止输出有效电平。

<3> 若 CR5n 与计数值相等，将输出非有效电平直至溢出再次发生。

<4> 重复 <2>和 <3>的操作，直至计数停止。

<5> 当 TCE5n=0 停止计数时，PWM 输出非有效电平。

需要了解时序的详细信息，参见图 **7-14** 和 **7-15**。

周期、有效电平宽度和占空比如下。

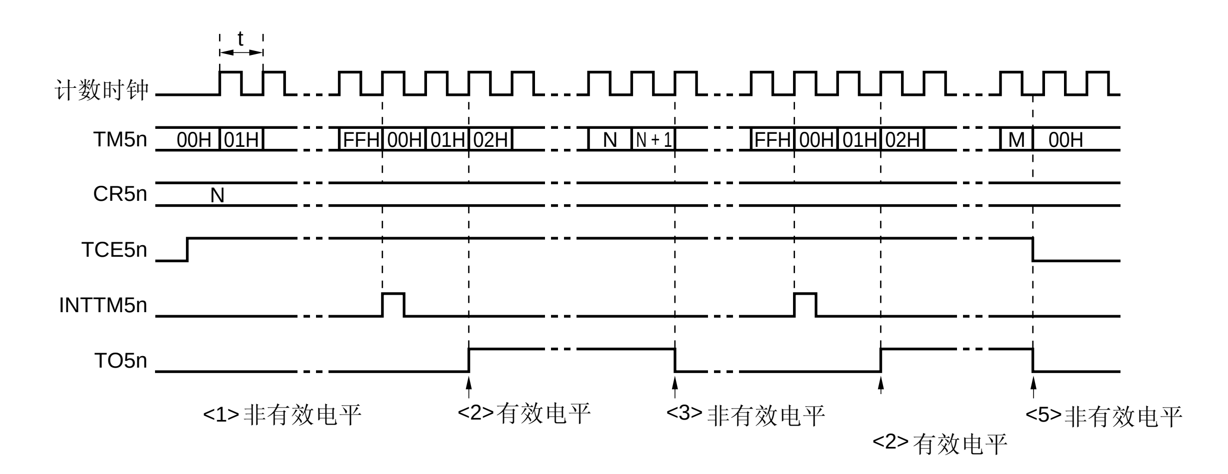
- 周期 = $2^8$t
- 有效电平宽度 = Nt
- 占空比 = N/$2^8$

  (N = 00H ～ FFH)

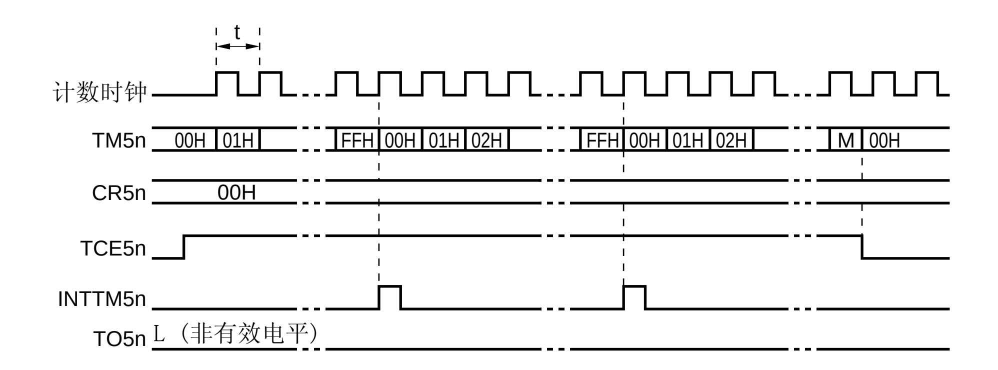
备注 n = 0, 1

**图 7-14. PWM 输出操作的时序**

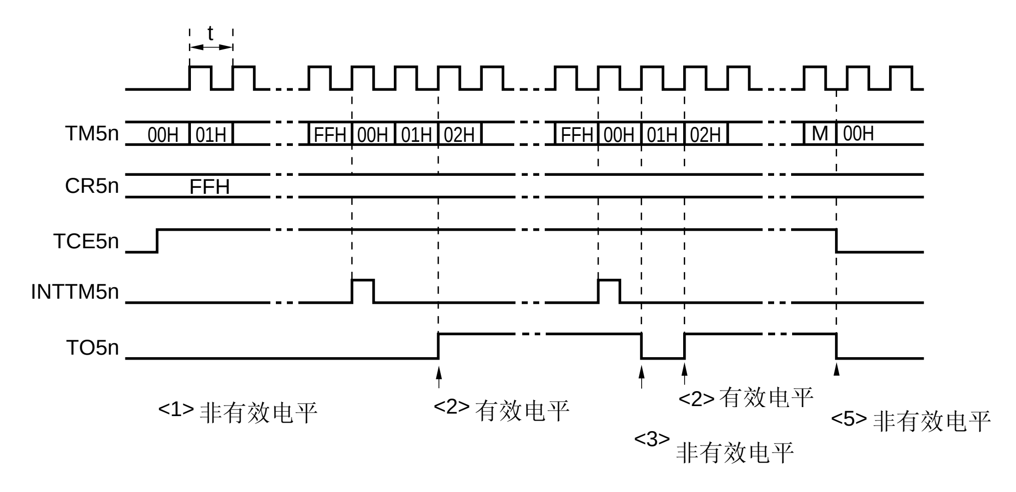
**(a) 基本操作 (有效电平 = H)**




**(b) CR5n = 00H**




**(c) CR5n = FFH**



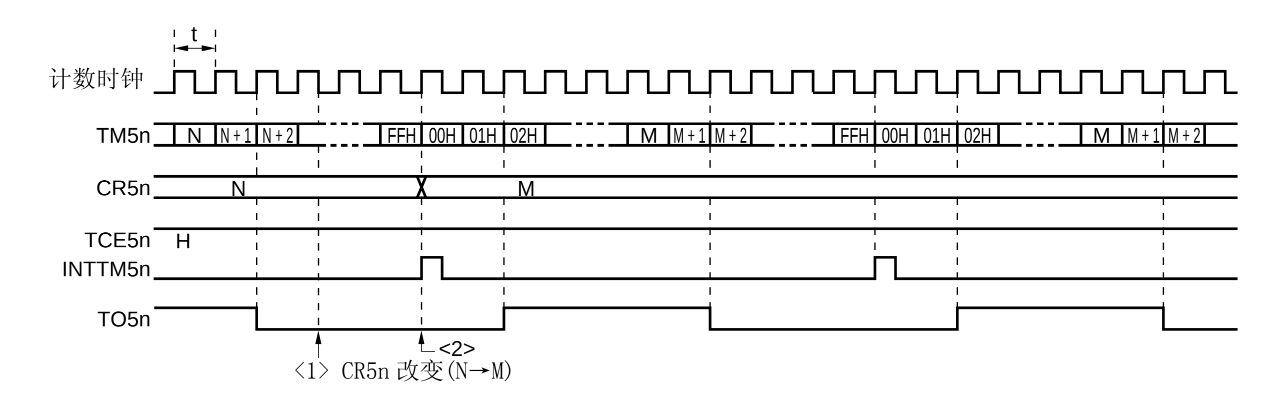

备注 **1.** 图 7-14 (a)中的<1> ～ <3> 和 <5> 与 **7.4.4 (1) PWM 输出基本操作**的 PWM 输出操作 中<1> ～ <3> 和 <5>相对应。

**2.** n = 0, 1

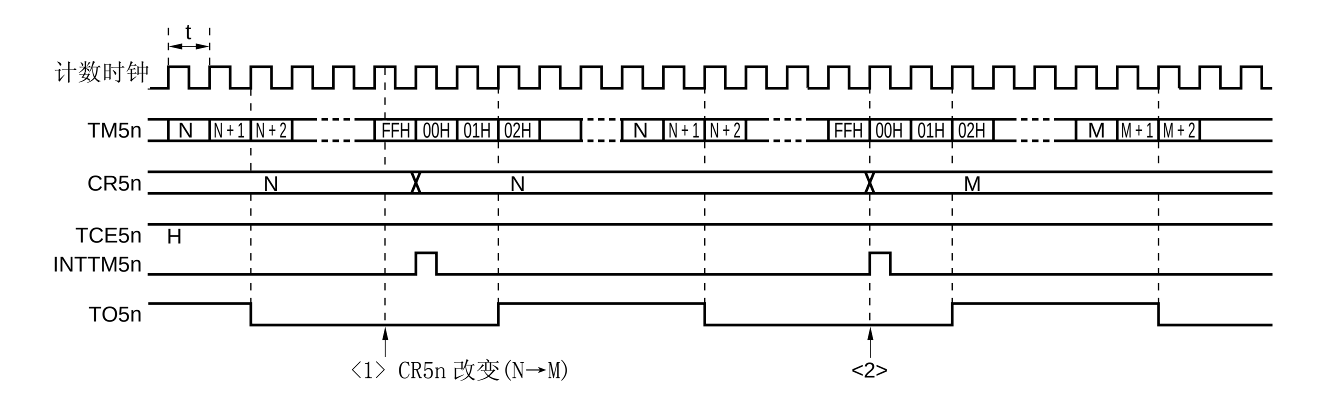
**(2)** 当 **CR5n** 改变时的操作

图 **7-15.** 改变 **CR5n** 时的操作时序

**(a)** 在计数值为 **FFH** 时，时钟上升沿出现之前，**CR5n** 的值由 **N** 变为 **M**
→改变后，溢出时立即将值传送至 **CR5n**。




**(b)** 在计数值为 **FFH** 时，时钟上升沿出现之后，**CR5n** 的值由 **N** 变为 **M**
→ 当出现第 **2** 次溢出时将值传送至 **CR5n**。




注意事项　图 **7-15<1>**与**<2>**之间读取 **CR5n** 时，读取的值与实际值不同**(**读取的值：**M**，**CR5n** 实际值：**N)**。

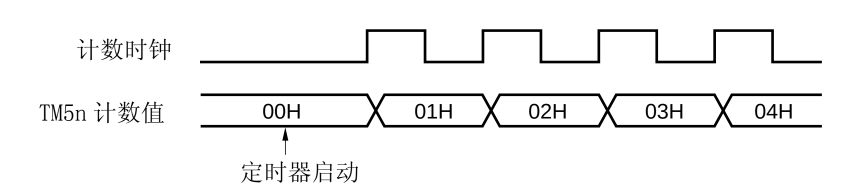
## 7.5 使用 8 位定时器/事件计数器 50 和 51 的注意事项

**(1)** 定时器启动误差

定时器启动后，在产生相等信号之前可能发生将近一个时钟周期的误差。

这是由于 8 位定时器计数器 50 和 51(TM50, TM51)的启动与计数时钟不同步引起的。

**图 7-16. 8 位定时器计数器 5n 启动的时序**




备注　　n = 0, 1

# 第八章　8 位定时器 H0 和 H1

## 8.1  8 位定时器 H0 和 H1 的功能

8 位定时器 H0 和 H1 的功能如下。

- 间隔定时器
- 方波输出
- PWM 输出模式
- 载波发生器模式 (仅用于 8 位定时器 H1)

## 8.2  8 位定时器 H0 和 H1 的配置

8 位定时器 H0 和 H1 包括以下硬件

**表 8-1.  8 位定时器 H0 和 H1 的配置**

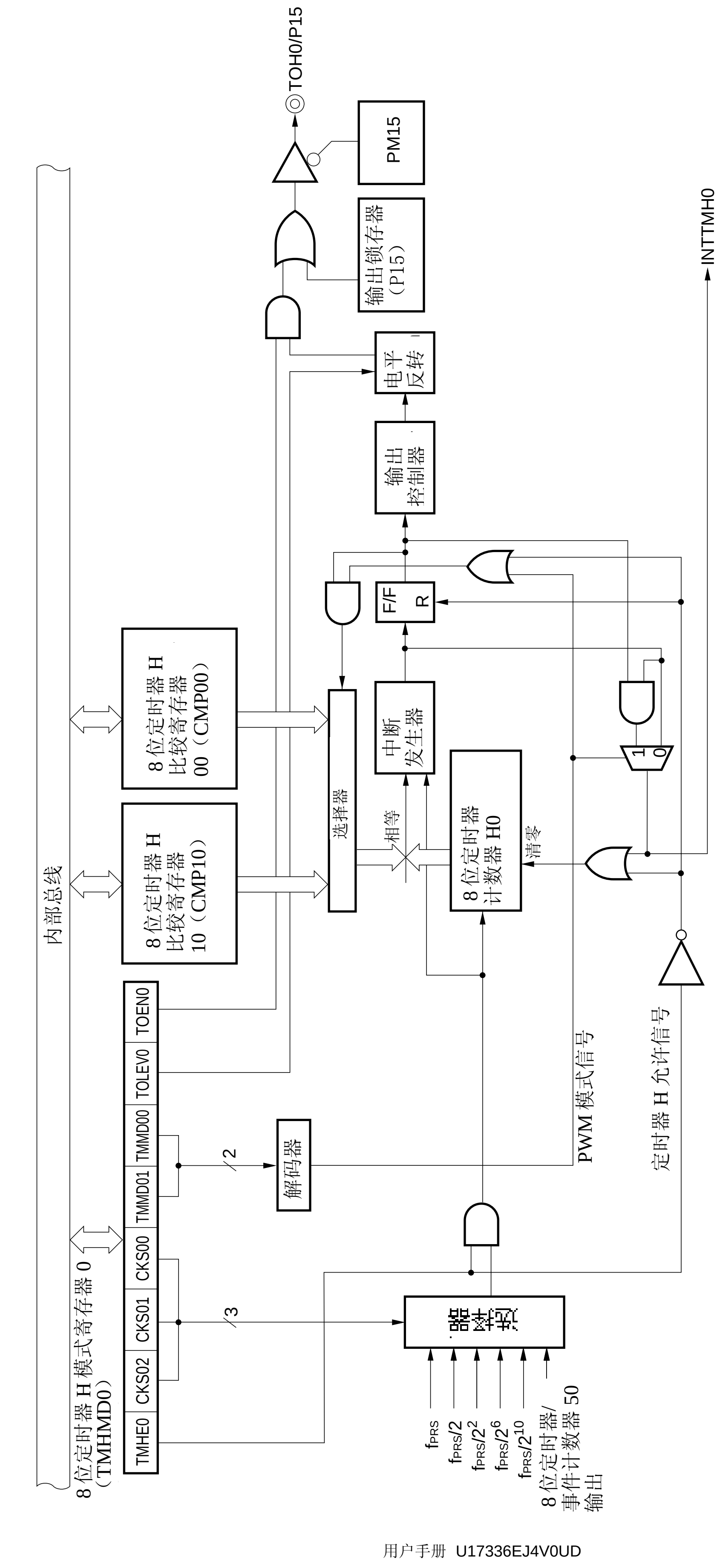
| 项目 | 配置 |
|---|---|
| 定时器寄存器 | 8 位定时器计数 Hn |
| 寄存器 | 8 位定时器 H 比较寄存器 0n （CMP0n）<br>8 位定时器 H 比较寄存器 1n （CMP1n））|
| 定时器输出 | TOHn，输出控制器 |
| 控制寄存器 | 8 位定时器H模式寄存器n （TMHMDn）<br>8 位定时器 H 载波控制寄存器 1 （TMCYC1）[注]<br>端口模式寄存器 1 （PM1）<br>端口寄存器 1 （P1） |

**注**　　仅用于 8 位定时器 H1

**备注**　　n = 0, 1

图 8-1 和 8-2 为定时器框图。

**图 8-1. 8 位定时器 H0 的框图**



• 图 8-2. 8 位定时器 H1 的框图

内部总线
8 位定时器模式控制
寄存器 1（TMHMD1）
TMHE1
CKS12
CKS11
CKS10
TMMD11
TMMD10
TOLEV1
TOEN1
8 位定时器 H
比较寄存器 11
（CMP11）
8 位定时器 H
比较寄存器 01
（CMP01）
RMC1
NRZB1
NRZ1
8 位定时器 H 载波
控制寄存器 1
（TMCYC1）
重载/
中断控制
INTTM51
3
2
解码器
选择器
TOH1/
INTP5/
P16
相等
中断
发生器
F/F
R
输出
控制器
电平
反转
输出锁存器
（P16）
PM16
$f_{PRS}$
$f_{PRS}/2^2$
$f_{PRS}/2^4$
$f_{PRS}/2^6$
$f_{PRS}/2^{12}$
$f_{RL}$
$f_{RL}/2^7$
$f_{RL}/2^9$
选择器
8 位定时器
计数器 H1
载波发生器模式信号
清零
PWM 模式信号
1
0
定时器 H 允许信号
INTTMH1

**(1)　8 位定时器 H 比较寄存器 0n (CMP0n)**

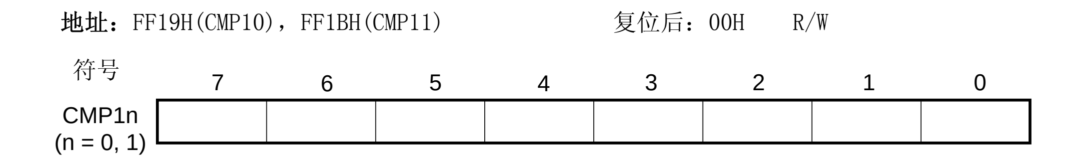
该寄存器可由 8 位存储器操作指令进行读写。该寄存器用于所有的定时器操作模式。

该寄存器不断地将 CMP0n 的值和 8 位定时器计数器 Hn 的计数值进比较，并且当两个值相等时，产生一个中断请求信号（INTTMHn）并反转 TOHn 的输出。

当定时器停止时(TMHEn = 0)重写 CMP0n 的值。

复位信号的产生将该寄存器设置为 00H。

**图 8-3.　8 位定时器 H 比较寄存器 0n（CMP0n）的格式**




<R>　　**注意事项　CMP0n 在定时器计数操作期间不能被重写。（只能更新与原数据相同的值）。**

**(2)　8 位定时器 H 比较寄存器 1n (CMP1n)**

可由 8 位存储器操作指令读写该寄存器。该寄存器用于 PWM 输出模式和载波发生器模式。

在 PWM 输出模式中，该寄存器不断地将 CMP1n 的设置值和 8 位定时器计数器 Hn 的计数值进行比较，当两个值相等时，反转 TOHn 的输出。不产生中断请求信号。

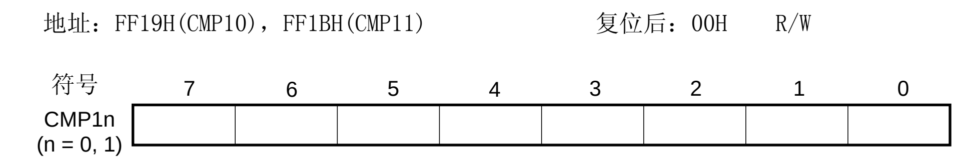
在载波发生器模式中，CMP1n 寄存器不断地将 CMP1n 的设置值和 8 位定时器计数器 Hn 的计数值进行比较，如果两个值相等，则产生一个中断请求信号（INTTMHn）。同时，计数值清零。

<R>　　可以在定时器计数期间对 CMP1n 进行更新（相同值）和重写。

如果在定时器操作期间重写 CMP1n 的值，则当定时器计数器的值和 CMP1n 以前的值相相等时，锁存新的值并传输给 CMP1n，然后 CMP1n 的值用新值取代。如果计数值和 CMP1n 值的相等操作与 CMP1n 的写操作发生冲突，那么 CMP1n 的值将保持不变。

复位信号的产生对该寄存器清零(00H)。

**图 8-4.　8 位定时器 H 比较寄存器 1n 的格式(CMP1n)**




注意事项　**在 PWM 输出模式和载波发生器模式下，当停止定时器计数操作(TMHEn = 0)后再启动定时器计数操作(TMHEn = 1)时，必须设置 CMP1n（即使对 CMP1n 设置相同的值也要再设置一次）。**

备注　n = 0, 1

## 8.3 控制 8 位定时器 H0 和 H1 的寄存器

以下四个寄存器用于控制 8 位定时器 H0 和 H1。

- 8 位定时器 H 模式寄存器 n （TMHMDn）
- 8 位定时器H载波控制寄存器 1 （TMCYC1）[注]
- 端口模式寄存器 1（PM1）
- 端口寄存器 1（P1）

注 仅用于 8 位定时器 H1

**（1）8 位定时器 H 模式寄存器 n （TMHMDn）**

该寄存器用于控制定时器 H 的模式。

可以由 1 位或 8 位存储器操作指令设置该寄存器。

复位信号的产生对寄存器清零（00H）。

备注 n = 0, 1

**图 8-5. 8 位定时器 H 模式寄存器 0 （TMHMD0）的格式**

地址：FF69H　　　　　　复位后：00H　　R/W

| | <7> | 6 | 5 | 4 | 3 | 2 | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| TMHMD0 | TMHE0 | CKS02 | CKS01 | CKS00 | TMMD01 | TMMD00 | TOLEV0 | TOEN0 |

| TMHE0 | 是否允许定时器操作 |
|---|---|
| 0 | 停止定时器计数操作（计数器清零） |
| 1 | 允许定时器计数（由输入时钟启动计数操作） |

| CKS02 | CKS01 | CKS00 | 计数时钟选择 | | | | |
|---|---|---|---|---|---|---|---|
| | | | | $f_{PRS}$ = 2 MHz | $f_{PRS}$ = 5 MHz | $f_{PRS}$ = 10 MHz | $f_{PRS}$ = 20 MHz |
| 0 | 0 | 0 | $f_{PRS}$ | 2 MHz | 5 MHz | 10 MHz | 20 MHz |
| 0 | 0 | 1 | $f_{PRS}/2$ | 1 MHz | 2.5 MHz | 5 MHz | 10 MHz |
| 0 | 1 | 0 | $f_{PRS}/2^2$ | 500 kHz | 1.25 MHz | 2.5 MHz | 5 MHz |
| 0 | 1 | 1 | $f_{PRS}/2^6$ | 31.25 kHz | 78.13 kHz | 156.25 kHz | 312.5 kHz |
| 1 | 0 | 0 | $f_{PRS}/2^{10}$ | 1.95 kHz | 4.88 kHz | 9.77 kHz | 19.54 kHz |
| 1 | 0 | 1 | TM50 输出[注] | | | | |
| 其它情况 | | | 禁止设置 | | | | |

| TMMD01 | TMMD00 | 定时器操作模式 |
|---|---|---|
| 0 | 0 | 间隔定时器模式 |
| 1 | 0 | PWM 输出模式 |
| 其它情况 | | 禁止设置 |

| TOLEV0 | 定时器输出电平控制（默认模式） |
|---|---|
| 0 | 低电平 |
| 1 | 高电平 |

| TOEN0 | 定时器输出控制 |
|---|---|
| 0 | 禁止输出 |
| 1 | 允许输出 |

注　在选择 TM50 输出作为计数时钟时，应注意以下几点。

- TM50 和 CR50 的值相相等时产生清零&启动模式（TMC506 = 0），
  先启动 8 位定时器/事件计数器 50，然后允许定时器进行 F/F 反转操作（TMC501 = 1）
- PWM 模式（TMC506 = 1）
  先启动 8 位定时器/事件计数器 50，然后设置计数时钟，使占空比= 50%

在任何模式中都不必将 TO50 作为定时器输出引脚。

<R>　注意事项　**1.** 当 **TMHE0=1** 时，禁止设置 **TMHMD0** 的其他位。但是可以将 **TMHMD0** 更新（相同值）。

**2.** 在 **PWM** 输出模式中，当定时器停止计数后（**TMHE0 = 0**）再次启动计数操作时（**TMHE0 = 1**），必须设置 **8** 位定时器 **H** 比较寄存器 **10**（**CMP10**）的值（即使设置的是相同的值，也必须再次设置）。

备注　**1.** $f_{PRS}$: 外围硬件时钟频率

**2.** TMC506: 8 位定时器模式控制寄存器 50（TMC50）的第 6 位
TMC501: TMC50 的第 1 位

**图 8-6. 8 位定时器 H 模式寄存器 1 （TMHMD1）的格式**

地址：FF6CH　复位后：00H　R/W

| | <7> | 6 | 5 | 4 | 3 | 2 | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| TMHMD1 | TMHE1 | CKS12 | CKS11 | CKS10 | TMMD11 | TMMD10 | TOLEV1 | TOEN1 |

| TMHE1 | 是否允许定时器操作 |
|---|---|
| 0 | 停止定时器计数操作（计数器清零） |
| 1 | 允许定时器计数（由输入时钟启动计数操作） |

| CKS12 | CKS11 | CKS10 | 计数时钟选择 | | | | |
|---|---|---|---|---|---|---|---|
| | | | | $f_{PRS}$ = 2 MHz | $f_{PRS}$ = 5 MHz | $f_{PRS}$ = 10 MHz | $f_{PRS}$ = 20 MHz |
| 0 | 0 | 0 | $f_{PRS}$ | 2 MHz | 5 MHz | 10 MHz | 20 MHz |
| 0 | 0 | 1 | $f_{PRS}/2^2$ | 500 kHz | 1.25 MHz | 2.5 MHz | 5 MHz |
| 0 | 1 | 0 | $f_{PRS}/2^4$ | 125 kHz | 312.5 kHz | 625 kHz | 1.25 MHz |
| 0 | 1 | 1 | $f_{PRS}/2^6$ | 31.25 kHz | 78.13 kHz | 156.25 kHz | 312.5 kHz |
| 1 | 0 | 0 | $f_{PRS}/2^{12}$ | 0.49 kHz | 1.22 kHz | 2.44 kHz | 4.88 kHz |
| 1 | 0 | 1 | $f_{RL}/2^7$ | 1.88 kHz (TYP.) | | | |
| 1 | 1 | 0 | $f_{RL}/2^9$ | 0.47 kHz (TYP.) | | | |
| 1 | 1 | 1 | $f_{RL}$ | 240 kHz (TYP.) | | | |

| TMMD11 | TMMD10 | 定时器操作模式 |
|---|---|---|
| 0 | 0 | 间隔定时器模式 |
| 0 | 1 | 载波发生器模式 |
| 1 | 0 | PWM 输出模式 |
| 1 | 1 | 禁止设置 |

| TOLEV1 | 定时器输出电平控制（默认模式） |
|---|---|
| 0 | 低电平 |
| 1 | 高电平 |

| TOEN1 | 定时器输出控制 |
|---|---|
| 0 | 禁止输出 |
| 1 | 允许输出 |

<R> **注意事项**

1. 当 **TMHE1=1** 时，禁止设置 **TMHMD1** 的其他位。但是可以将 **TMHMD1** 更新（相同值）。
2. 在 **PWM** 输出模式和载波发生模式中，当定时器停止计数后（**TMHE1=0**）再次启动计数器操作时（**TMHE1=1**），必须设置 **8** 位定时器 **H** 比较寄存器 **11**（**CMP11**）的值（即使设置的是相同的值，也必须再次设置）。
3. 当使用载波发生器模式时，设置 **TMH1** 的计数时钟频率至少是 **TM51** 计数时钟频率的 **6** 倍。

**备注**

1. $f_{PRS}$：外围硬件时钟频率
2. $f_{RL}$：内部低速振荡时钟频率

**(2) 8 位定时器 H 载波控制寄存器 1 （TMCYC1）**

该寄存器用于控制 8 位定时器 H1 的遥控输出和载波脉冲输出状态。

可以由 1 位或 8 位存储器操作指令设置该寄存器。

复位信号的产生对寄存器清零（00H）。

**图 8-7.　8 位定时器 H 载波控制寄存器 1（TMCYC1）的格式**

地址：FF6DH　复位后：00H　R/W[注]

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| TMCYC1 | 0 | 0 | 0 | 0 | 0 | RMC1 | NRZB1 | NRZ1 |

| | RMC1 | NRZB1 | 遥控输出 |
|---|---|---|---|
| | 0 | 0 | 低电平输出 |
| <R> | 0 | 1 | 在 INTTM51 信号输入的上升沿高电平输出 |
| | 1 | 0 | 低电平输出 |
| <R> | 1 | 1 | 在 INTTM51 信号输入的上升沿载波脉冲输出 |

| NRZ1 | 载波脉冲输出状态标志 |
|---|---|
| 0 | 禁止载波输出状态（低电平状态） |
| 1 | 允许载波输出状态<br>（RMC1=1: 载波脉冲输出，RMC1=0: 高电平状态 |

注　第 0 位只读。

<R>　**注意事项　当 TMHE=1 时，禁止重写 RMC1。但是可以将 TMCYC1 更新（相同值）。**

**(3) 端口模式寄存器 1（PM1）**

该寄存器按位设置端口 1 为输入输出操作模式。

在将 P15/TOH0 和 P16/TOH1/INTP5 引脚用于定时器输出时，需对 PM15 和 PM16 以及 P15 和 P16 的输出锁存器清零(0)。

可以由 1 位或 8 位存储器操作指令设置 PM1。

复位信号的产生将寄存器置 FFH。

**图 8-8.　端口模式寄存器 1（PM1）的格式**

地址: FF21H　复位后: FFH　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM1 | PM17 | PM16 | PM15 | PM14 | PM13 | PM12 | PM11 | PM10 |

| PM1n | P1n 引脚 I/O 模式 选择 (n = 0 ～ 7) |
|---|---|
| 0 | 输出模式（输出缓冲器打开） |
| 1 | 输入模式 (输出缓冲器关闭) |

## 8.4 8 位定时器 H0 和 H1 的操作

### 8.4.1 间隔定时器/方波输出操作

当 8 位定时器计数器 Hn 与比较寄存器 0n（CMP0n）相等时，将产生中断请求信号（INTTMHn）并对 8 位定时器计数器 Hn 清零（00H）。

在间隔定时器模式中不使用比较寄存器 1n（CMP1n）。即使 CMP1n 寄存器已赋值，也不会检测 8 位定时器计数器 Hn 与 CMP1n 寄存器的相等情况，所以不影响定时器的输出。

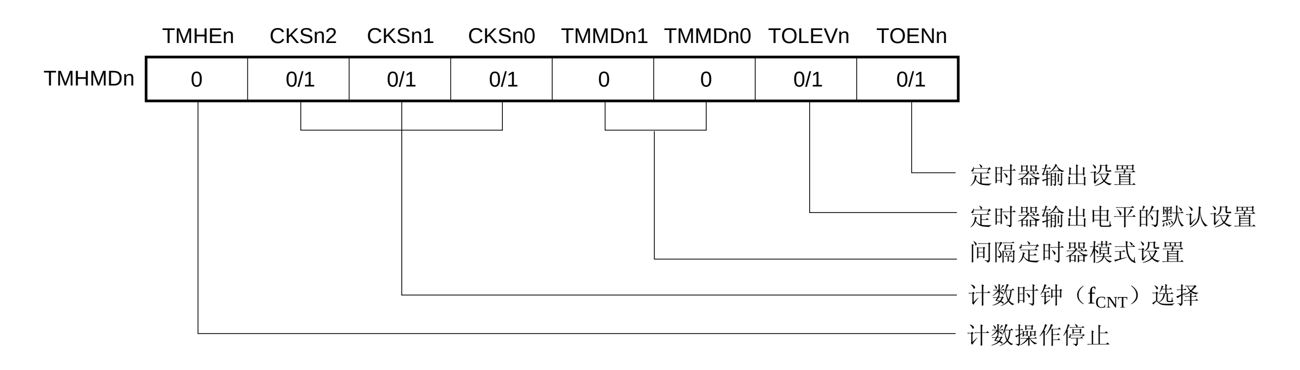
通过将定时器 H 模式寄存器 n（TMHMDn）的第 0 位（TOENn）置 1，可以从 TOHn 输出任意频率的方波（占空比 = 50%）。

设置

<1> 寄存器设置。

图 8-9. 间隔定时器/方波输出操作期间的寄存器设置

(i) 设置定时器 H 模式寄存器 n （TMHMDn）




(ii) CMP0n 寄存器设置

如果 N 作为比较值，则时间间隔如下：

- 时间间隔 = (N +1)/$f_{CNT}$

<2> 当 TMHEn = 1 时，开始计数。

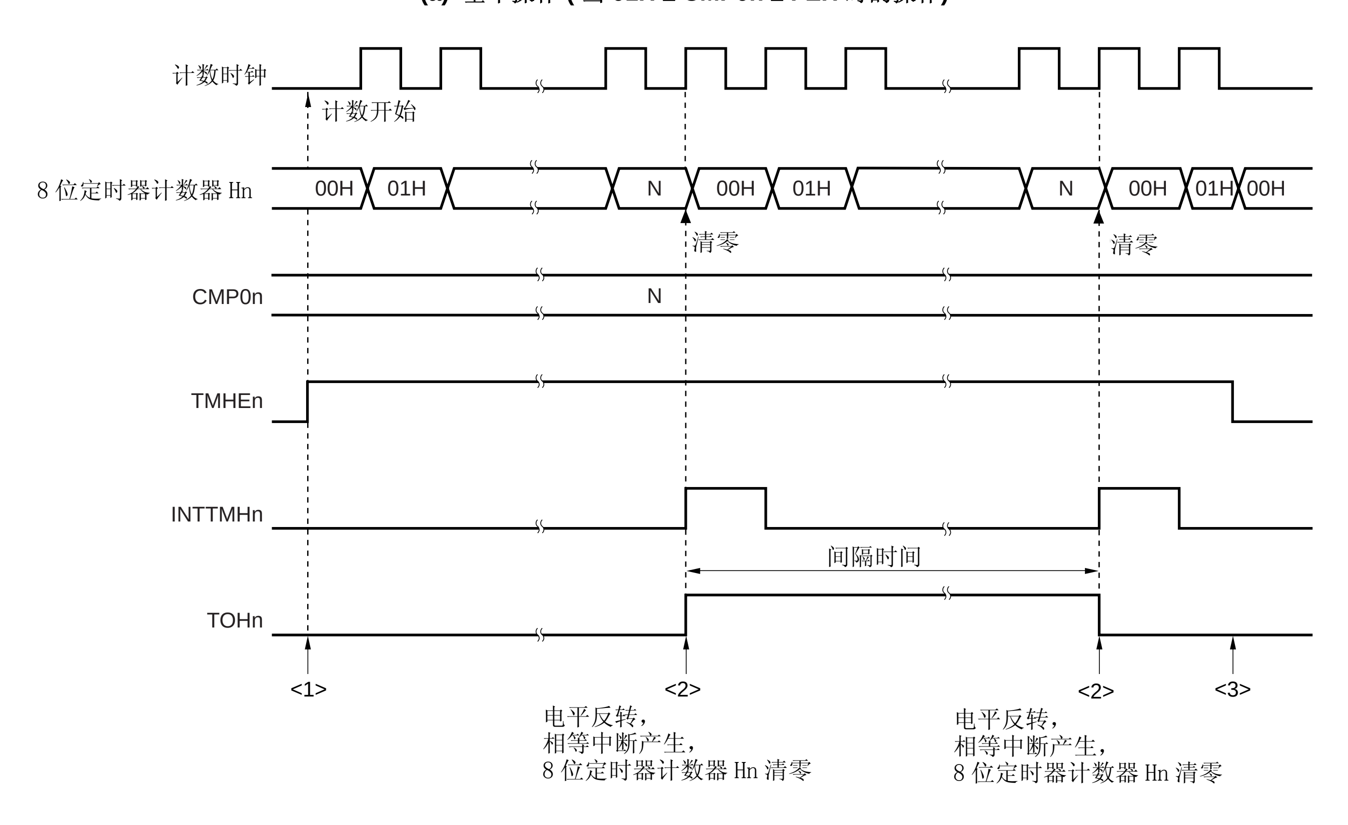
<3> 当 8 位定时器计数器 Hn 与 CMP0n 的值相等时，将产生 INTTMHn 信号，并对 8 位定时器计数器 Hn 清零（00H）。

<4> 随后，以相同的间隔时间重复产生 INTTMHn 信号。若要停止计数操作，则将 TMHEn 清零(0)。

备注 **1.** 如需了解输出引脚的设置情况，参见 **8.3 (3) 端口模式寄存器 1（PM1）。**
**2.** 如需了解允许 INTTMHn 信号中断的情况，可参见 **第十八章 中断功能。**
**3.** n = 0, 1

图 8-10. 间隔定时器/方波输出操作时序（1/2）

(a) 基本操作 ( 当 01H ≤ CMP0n ≤ FEH 时的操作)



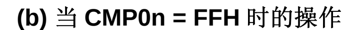

<1> 通过将 TMHEn 置 1，允许计数操作。一旦允许计数，则在不到 1 个时钟周期的时间内启动计数操作。

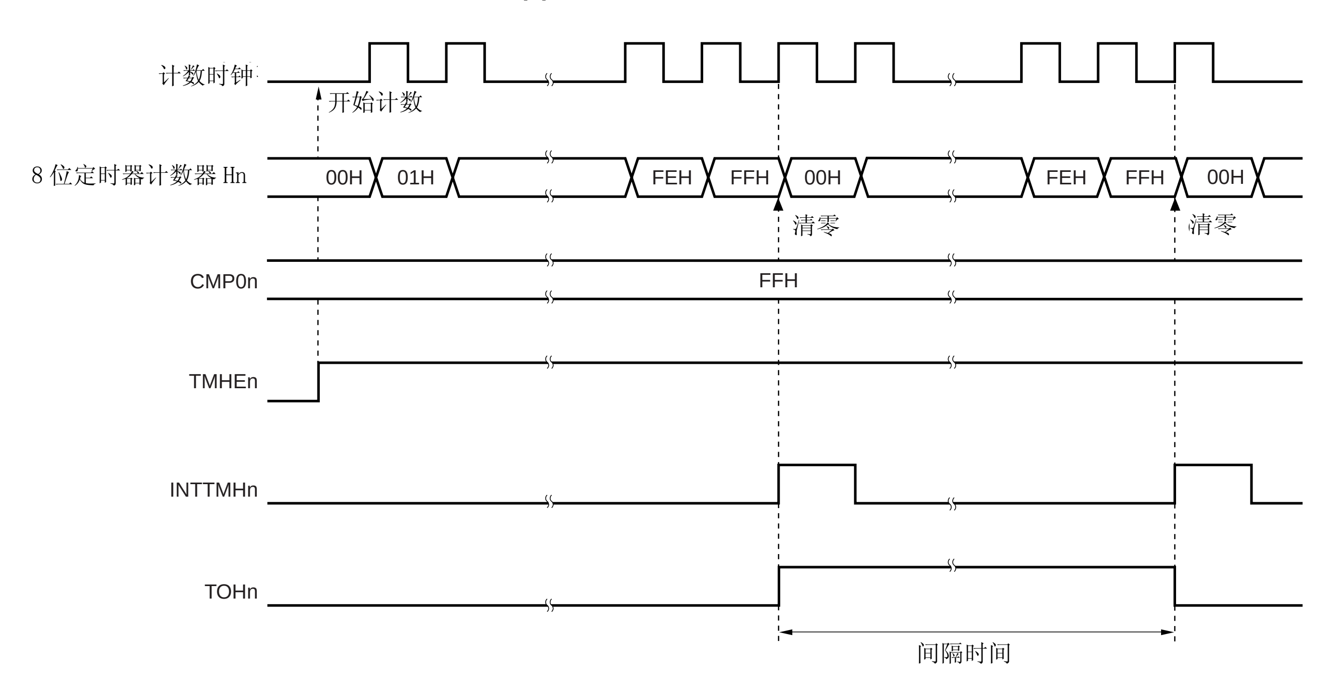
<2> 当 8 位定时器计数器 Hn 与 CMP0n 寄存器的值相等时，将该定时器计数器清零并反转 TOHn 的输出电平。此外在计数时钟的上升沿输出 INTTMHn 信号。

<3> 在定时器 Hn 操作期间将 TMHEn 位清零，设置 INTTMHn 信号和 TOHn 的输出为默认电平。如果在将 TMHEn 位清零(0)前它们已经处于默认电平，则电平保持不变。

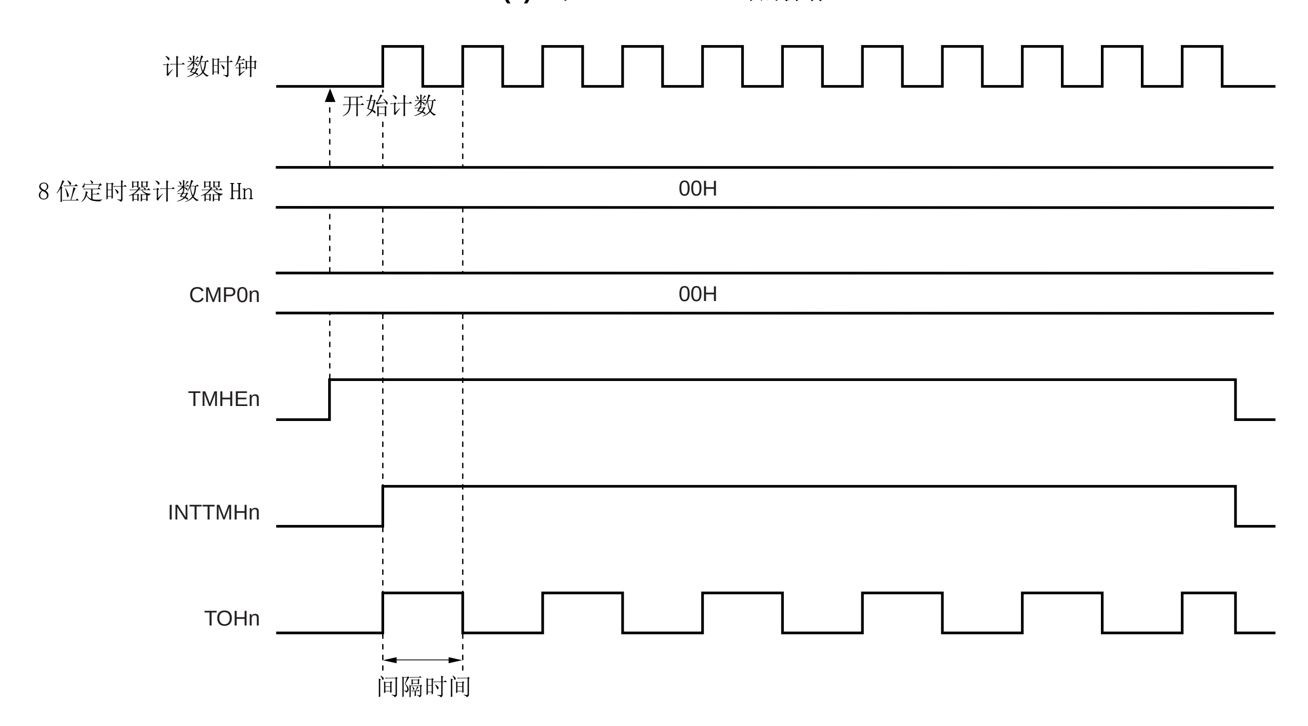
备注　n = 0, 1
　　　01H ≤ N ≤ FEH

图 **8-10.** 间隔定时器/方波输出操作时序 （**2/2**）




**(c)** 当 **CMP0n = 00H** 时的操作




**备注** n = 0, 1

### 8.4.2 PWM 输出操作

在 PWM 输出模式中，可以输出具有任意占空比和周期的脉冲。

8 位定时器比较寄存器 0n（CMP0n）用于控制定时器输出（TOHn）的周期。禁止在定时器操作期间修改 CMP0n 寄存器的值。

8 位定时器比较寄存器 1n（CMP1n）用于控制定时器输出（TOHn）的占空比。允许在定时器操作期间修改 CMP1n 寄存器的值。

PWM 输出模式中的操作如下所示。

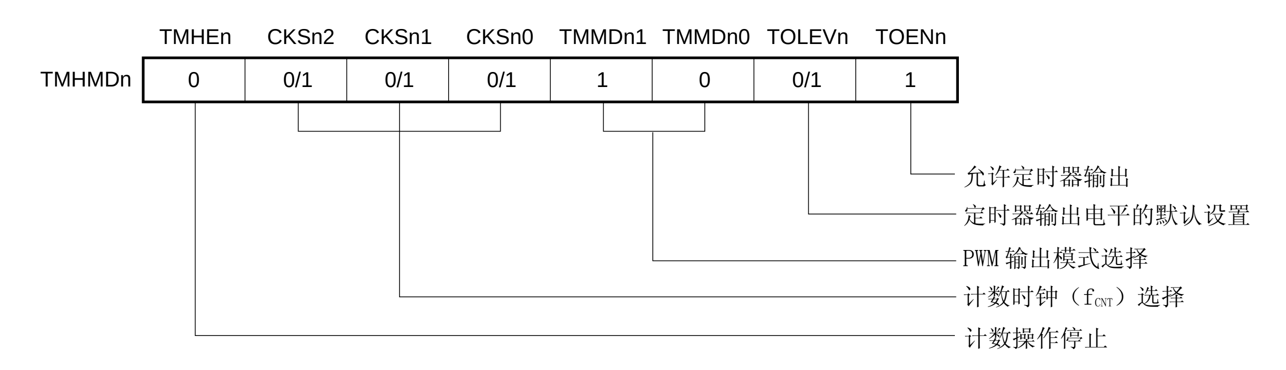
在定时器开始计数后，当 8 位定时器计数器 Hn 与 CMP0n 寄存器的值相等时，反转 TOHn 的输出电平且清零 8 位定时器计数器 Hn。当 8 位定时器计数器 Hn 与 CMP1n 寄存器的值相等时，反转 TOHn 的输出电平。

设置

<1>　寄存器设置。

**图 8-11. PWM 输出模式下的寄存器设置**

**(i)　设置定时器 H 模式寄存器 n (TMHMDn)**




**(ii)　设置寄存器 CMP0n**

- 比较值(N): 周期设置

**(iii)　设置寄存器 CMP1n**

- 比较值(M): 占空比设置

备注　**1.** n = 0, 1

　　　**2.** 00H ≤ CMP1n (M) < CMP0n (N) ≤ FFH

<2>　当 TMHEn = 1 时，开始计数。

<3>　CMP0n 是在允许计数操作后首次被比较的比较寄存器。当 8 位定时器计数器 Hn 与 CMP0n 寄存器的值相等时，将 8 位定时器计数器 Hn 清零、产生中断请求信号（INTTMHn），并且反转 TOHn 输出电平。同时切换比较寄存器（与 8 位定时器比较的寄存器）：由 CMP0n 改为 CMP1n 。

<4>　当 8 位定时器计数器 Hn 与 CMP1n 寄存器相等时，反转 TOHn 输出电平，并切换比较寄存器（与 8 位定时器比较的寄存器）：由 CMP1n 改为 CMP0n 。此时不对 8 位定时器计数器 Hn 清零，也不产生 INTTMHn 信号。

<5> 重复执行过程<3> 和 <4>，可以获取具有任意占空比的脉冲。
<6> 若要停止计数操作，则设置 TMHEn = 0。
若 CMP0n 的值为 N、CMP1n 的值为 M，计数时钟频率为 $f_{CNT}$，则 PWM 脉冲的输出周期和占空比如下所示。

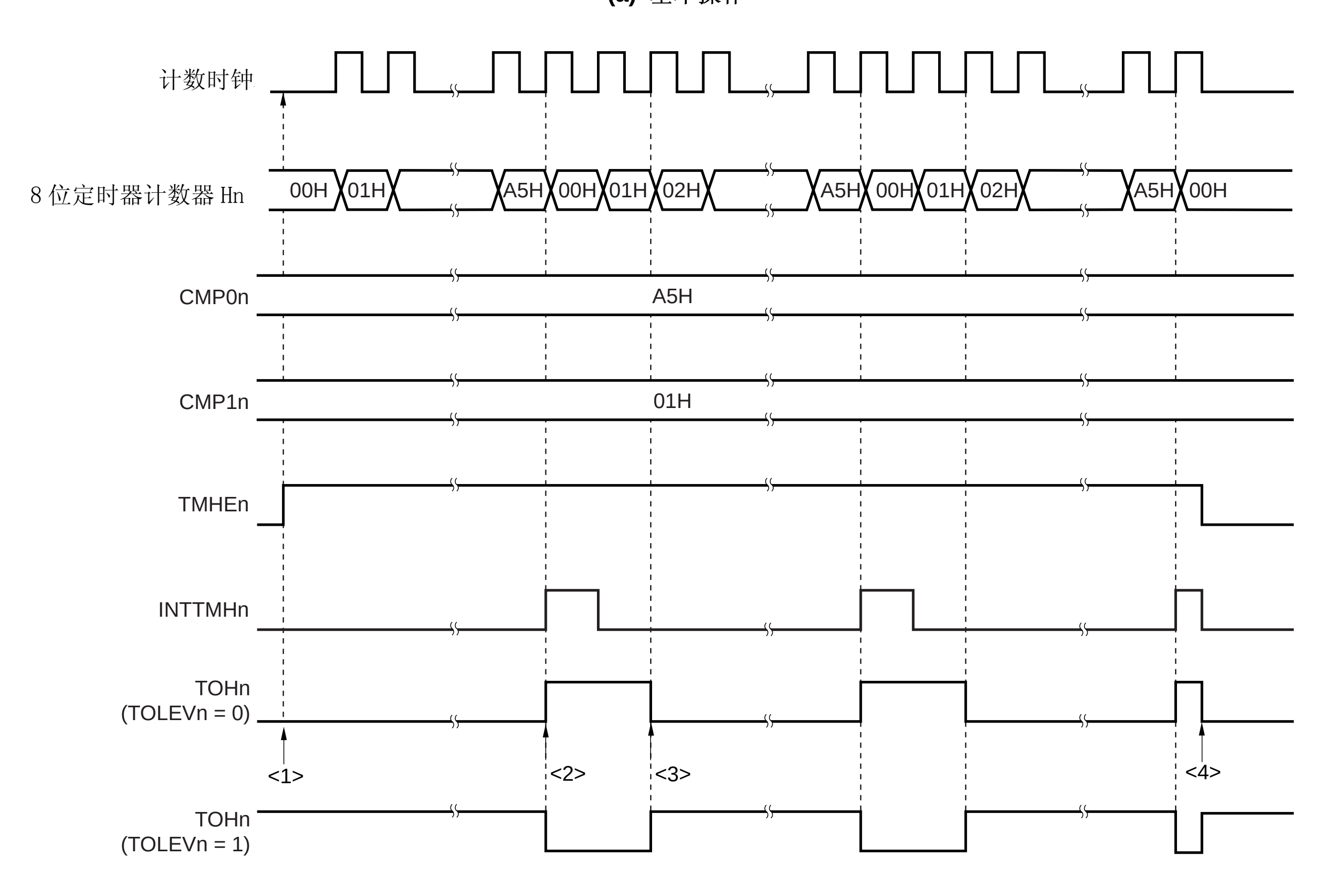
- PWM 脉冲输出周期 = $(N + 1)/f_{CNT}$
- 占空比 = $(M + 1)/(N + 1)$

注意事项 **1. 可以在定时器计数操作时改变 CMP1n 的设置值。但从改变 CMP1n 寄存器的值到这个值被传输给寄存器需要经历三个操作时钟（通过 TMHMDn 寄存器的 CKSn2 到 CKSn0 位选择的时钟信号）**
**2. 在定时器计数操作停止后（TMHEn=0），再次启动定时器操作时（TMHEn=1）,必须对 CMP1n 进行设置（即使是设置相同的值，也必须重新设置 CMP1n 寄存器）**
**3. CMP1n 寄存器的设置值（M）和 CMP0n 寄存器的设置值（N）必须在以下范围内取值。**
**00H ≤ CMP1n (M) < CMP0n (N) ≤ FFH**

备注 **1.** 如需了解输出引脚的设置，参见 **8.3 (3) 端口模式寄存器 1 (PM1)**。
**2**. 要了解如何允许 INTTMHn 信号中断的情况，可参见 **第十八章 中断功能**。
**3.** n = 0, 1

图 8-12. PWM 输出模式下的操作时序（1/4）

(a) 基本操作




<1> 通过设置 TMHEn =1，允许计数操作。通过屏蔽一个计数时钟，启动 8 位定时器计数器 Hn。此时，TOHn 输出保持默认电平状态。

<2> 当 8 位定时器计数器 Hn 与 CMP0n 相等时将 TOHn 的输出电平反转，对 8 位定时器计数器 Hn 清零，并输出 INTTMHn 中断信号。

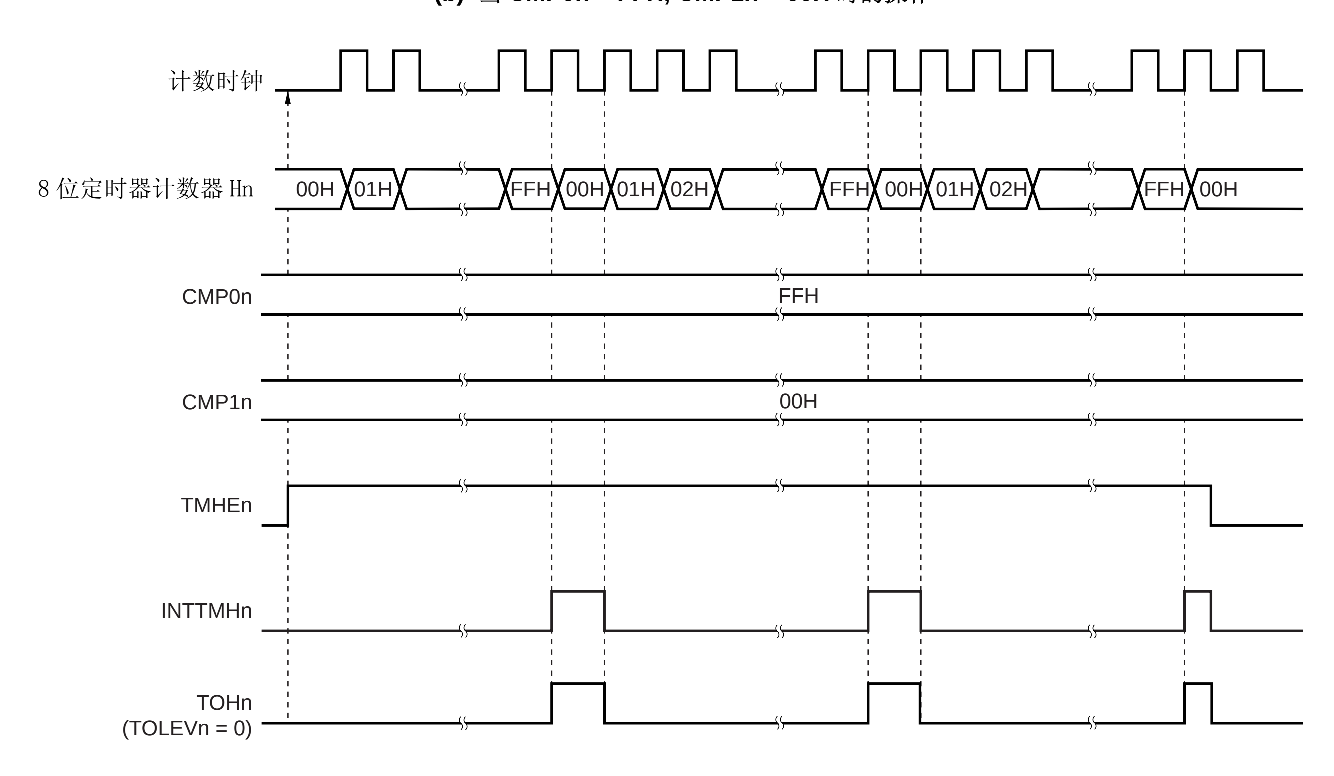
<3> 当 8 位定时器计数器 Hn 与 CMP1n 相等时，反转 TOHn 的输出电平。而此时并不对 8 位定时器计数器 Hn 清零，也不输出 INTTMHn 中断信号。

<4> 在定时器 Hn 操作期间若将 TMHEn 位清零，可以设置 INTTMHn 信号和 TOHn 输出电平为默认值。

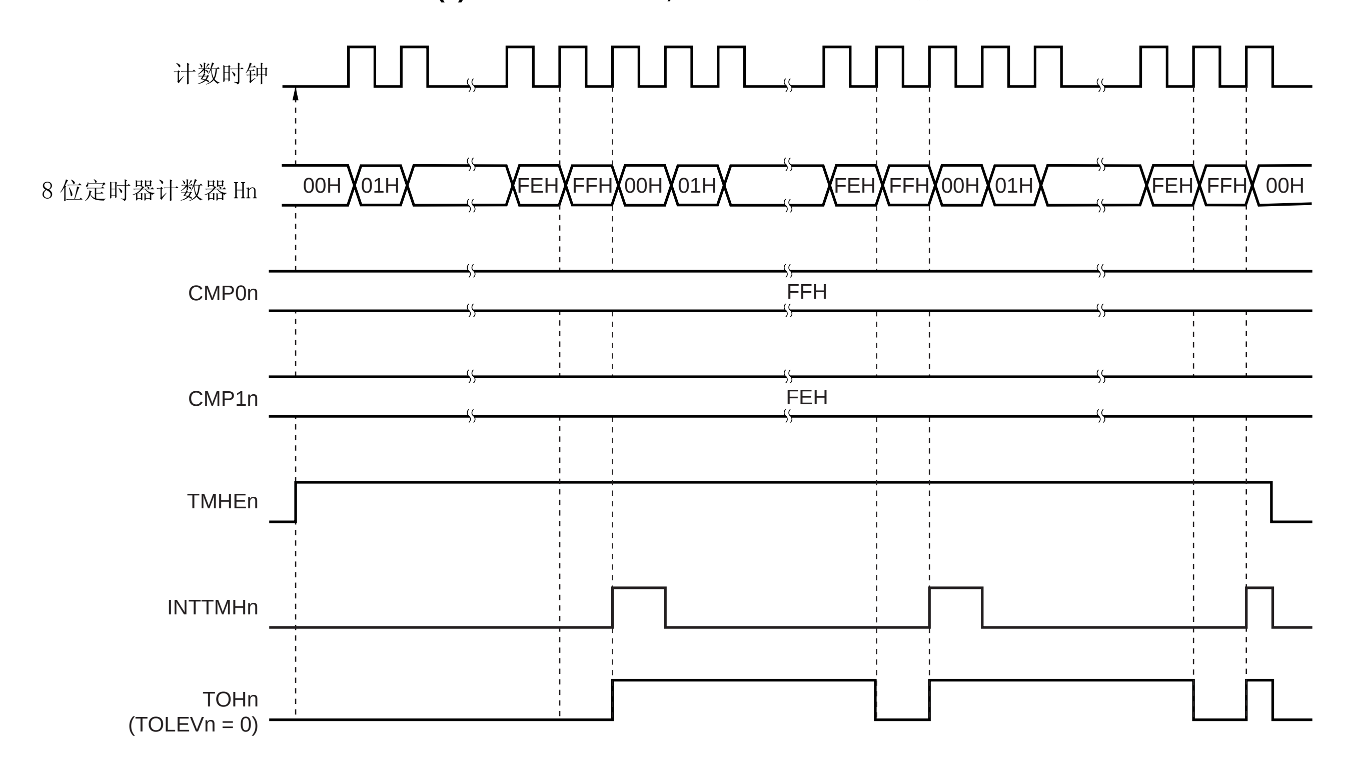
备注　　n = 0, 1

图 8-12. PWM 输出模式下的操作时序（2/4）

**(b) 当 CMP0n = FFH, CMP1n = 00H 时的操作**



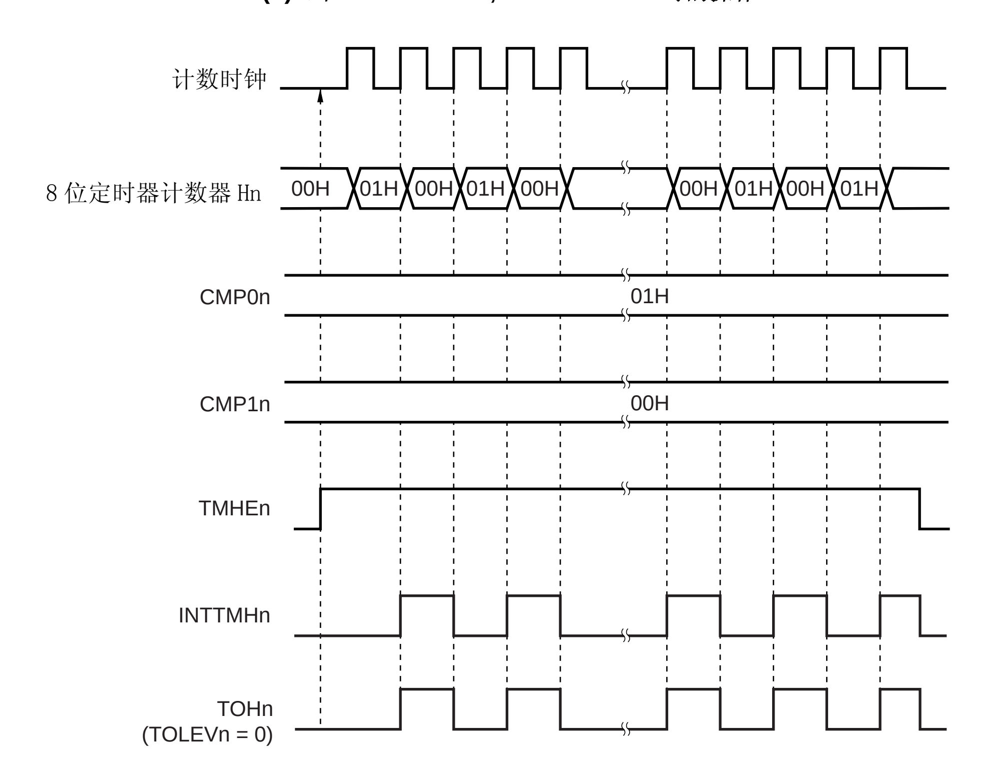

**(c) 当 CMP0n = FFH, CMP1n = FEH 时的操作**




备注　　n = 0, 1

**图 8-12.  PWM 输出模式下的操作时序（3/4）**

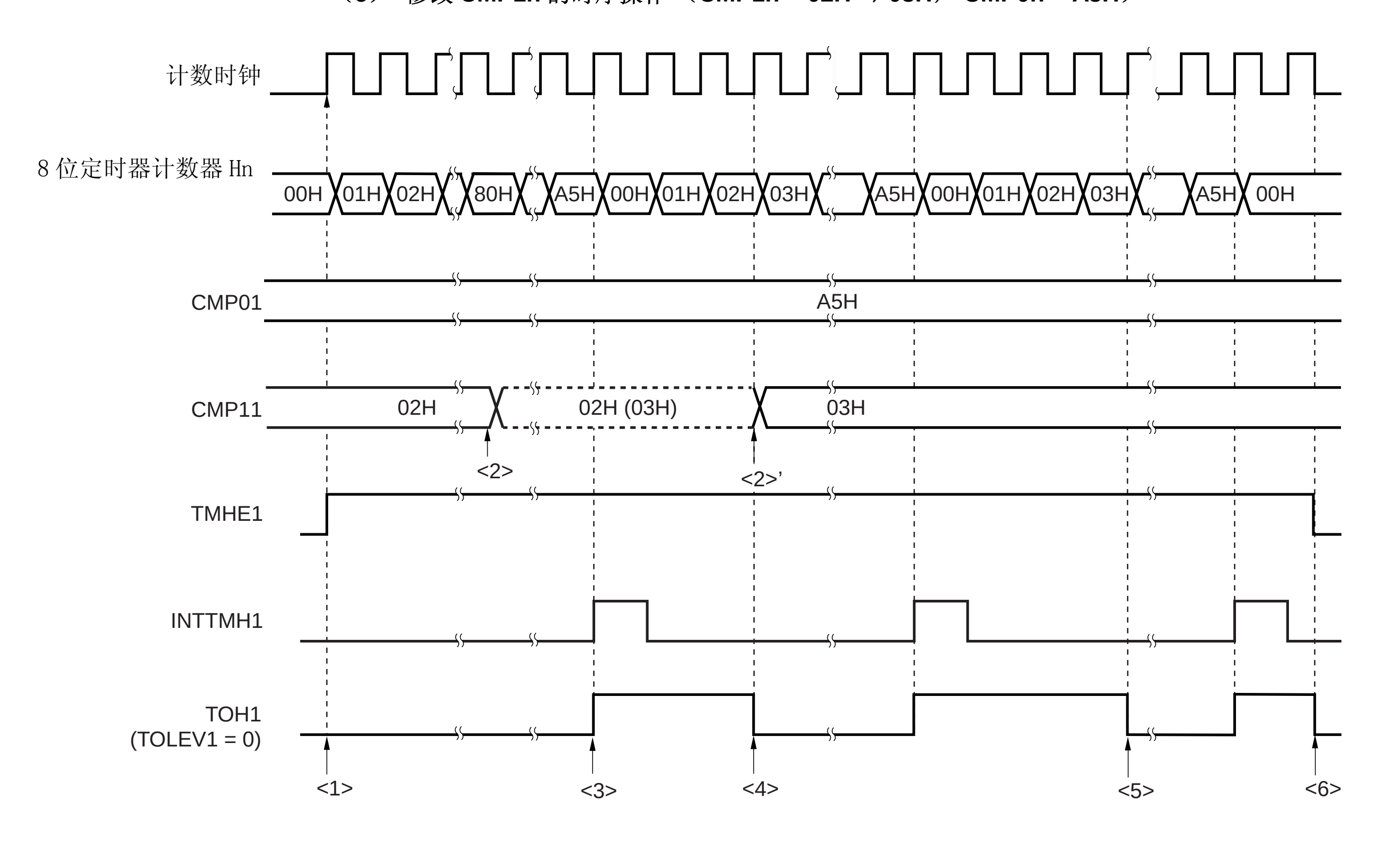
**(d)  当 CMP0n = 01H, CMP1n = 00H 时的操作**




备注　　n = 0, 1

**图 8-12. PWM 输出模式下的操作时序 （4/4）**

**（e） 修改 CMP1n 的时序操作 （CMP1n = 02H → 03H， CMP0n = A5H）**




<1> 通过设置 TMHEn =1 允许计数操作。通过屏蔽 1 个计数时钟，启动 8 位定时器计数器 Hn。此时 TOHn 输出保持默认电平状态。

<2> 在定时器计数器操作期间可以修改 CMP1n 的值。修改操作与计数时钟不同步。

<3> 当 8 位定时器计数器 Hn 与 CMP0n 的值相等时，将 8 位定时器计数器 Hn 的值清零、反转 TOHn 输出电平，并输出 INTTMHn 信号。

<4> 如果 CMP1n 的值被修改，则修改后的值被锁存，且不传送到寄存器。当 8 位定时器计数器 Hn 的值与修改前 CMP1n 寄存器的值相等时，则将值传送至 CMP1n，这样 CMP1n 的值就被修改了（<2>'）。
但从 CMP1n 的值被修改到将值传送至寄存器至少需要 3 个计数时钟。如果在 3 个计数时钟内产生一个相等信号，则不能将修改后的值传送至寄存器。

<5> 当 8 位定时器计数器 Hn 的值与修改后的 CMP1n 的值相等时，反转 TOHn 输出电平。此时不对 8 位定时器计数器 Hn 清零，也不产生 INTTMHn 信号。

<6> 在定时器 Hn 操作期间若将 TMHEn 位清零，可以设置 INTTMHn 信号和使 TOHn 输出电平为默认值。

**备注** n = 0, 1

### 8.4.3 载波发生器操作 （仅用于 8 位定时器 H1）

在载波发生器模式中 8 位定时器 H1 用于产生红外遥控器的载波信号，8 位定时器/事件计数器 51 用于红外遥控信号的产生（计时）

以 8 位定时器/事件计数器 51 设置的周期输出载波时钟。该载波时钟由 8 位定时器 H1 产生。

在载波发生器模式中，8 位定时器 H1 载波脉冲由 8 位定时器/事件计数器 51 控制输出，并从 TOH1 输出。

**(1)载波的发生**

在载波发生器模式中，8 位定时器 H 比较寄存器 01（CMP01）产生低电平宽度的载波脉冲波形，而 8 位定时器 H 比较寄存器 11（CMP11）产生高电平宽度的载波脉冲波形。

在 8 位定时器 H1 操作期间允许修改 CMP11，但禁止修改 CMP01。

**(2) 载波输出控制**

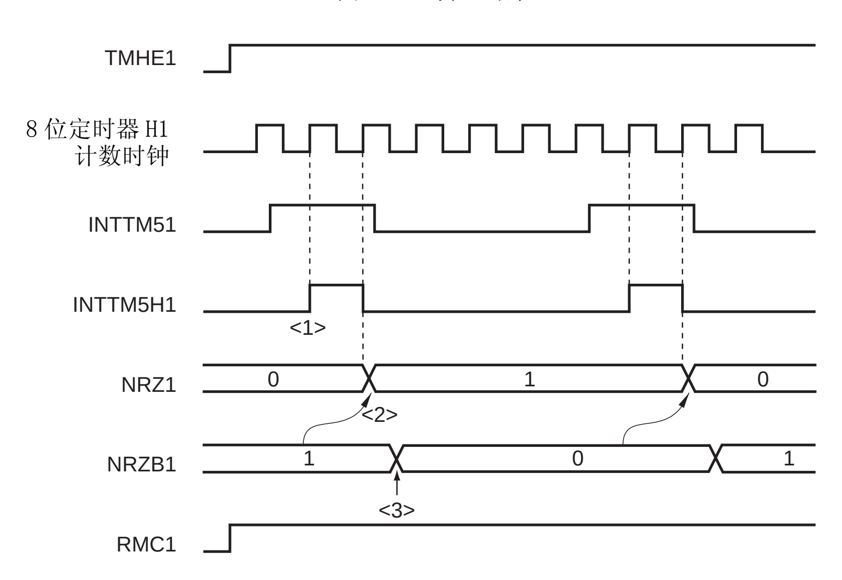
载波输出由 8 位定时器/事件计数器 51 的中断请求信号（INTTM51）和 8 位定时器 H 载波控制寄存器（TMCYC1）的 NRZB1 和 RMC1 位控制。输出之间的关系如下所示。

| RMC1 位 | NRZB1 位 | 输出 |
|---|---|---|
| 0 | 0 | 低电平输出 |
| 0 | 1 | 在 INTTM51 信号输入的上升沿， 高电平输出 |
| 1 | 0 | 低电平输出 |
| 1 | 1 | 在 INTTM51 信号输入的上升沿，载波脉冲输出 |

<R>
<R>

为了在计数操作期间控制载波脉冲输出，TMCYC1 寄存器的 NRZ1 和 NRZB1 位有 1 个主位和从位配置。NRZ1 位只读，而 NRZB1 位可读可写。INTTM51 信号与 8 位定时器 H1 计数时钟同步，且作为 INTTM5H1 信号输出。INTTM5H1 信号作为 NRZ1 位的数据传送信号，且 NRZB1 位的值被传送至 NRZ1 位。从 NRZB1 位传送至 NRZ1 位的时序如下所示。

图 **8-13.** 传送时序




<1> INTTM51 信号与 8 位定时器 H1 的计数时钟同步，并作为 INTTM5H1 信号输出。

<2> 在 INTTM5H1 信号上升沿的第 2 个时钟处将 NRZB1 位的值传送至 NRZ1 位。

<3> 在中断服务程序中将下一个值写入 NRZB1 位。该中断服务程序已由 INTTM5H1 中断启动或通过轮询中断请求标志检测到时序后启动。将下一次要计数的值写入 CR51 中。

注意事项 **1. NRZB1 位被重写后至少到第 2 个时钟时才能被再次重写，否则，从 NRZB1 位到 NRZ1 位的传送不能得到保证。**

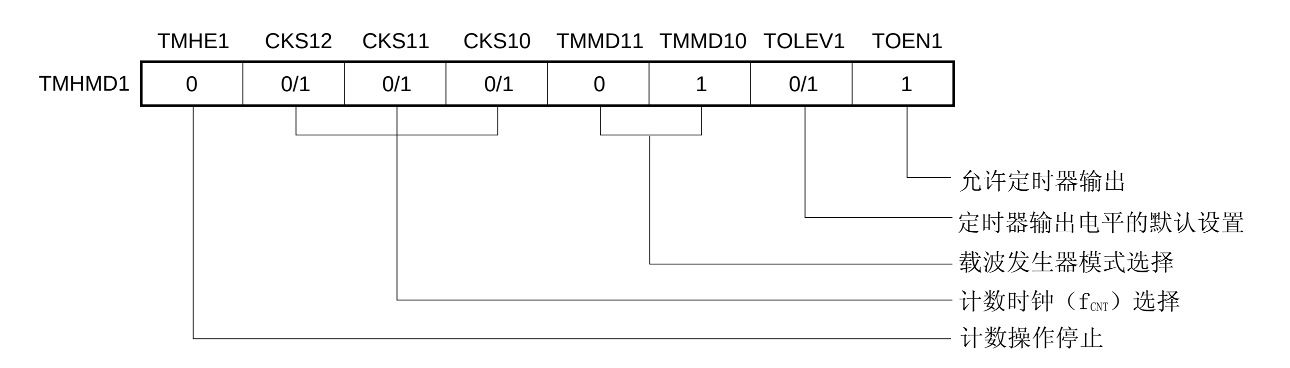
**2. 在载波发生器模式中使用 8 位定时器/事件计数器 51 时，在<1>时刻将产生中断。如果在其他模式中使用 8 位定时器/事件计数器 51，中断产生的时序则不相同。**

设置

<1>　寄存器设置

**图 8-14. 在载波发生器模式下的寄存器设置**

**(i)　设置 8 位定时器 H 模式寄存器 1 （TMHMD1）**




**(ii)　CMP01** 寄存器设置

- 比较值

**(iii)　CMP11** 寄存器设置

- 比较值

**(iv)　TMCYC1** 寄存器设置

- RMC1 = 1 ... 遥控输出允许位
- NRZB1 = 0/1 ... 载波输出允许位

**（v）　TCL51** 和 **TMC51** 寄存器设置

- 参见　**7.3　控制 8 位定时器/事件计数器 50 和 51 的寄存器**。

<2>　当 TMHE1 = 1 时，8 位定时器 H1 开始计数。

<3>　当 8 位定时器模式控制寄存器 51（TMC51）的 TCE51=1 时，8 位定时器/事件计数器 51 开始计数。

<4>　允许计数后，第 1 个用于比较的比较寄存器是 CMP01。当 8 位定时器计数器 H1 的计数值与 CMP01 的值相等时，将产生 INTTMH1 信号，并对 8 位定时器计数器 H1 清零，而此时与 8 位定时器计数器比较的比较寄存器从 CMP01 被切换为 CMP11。

<5>　当 8 位定时器计数器 H1 的计数值与 CMP11 寄存器的值相等时，将产生 INTTMH1 信号，并对 8 位定时器计数器 H1 清零，而此时与 8 位定时器计数器比较的比较寄存器从 CMP11 被切换为 CMP01。

<6>　重复执行过程<4>和<5>，将产生 1 个载波时钟。

<7>　INTTM51 信号与 8 位定时器 H1 的计数时钟同步，并作为 INTTM5H1 信号输出。INTTM5H1 信号作为 NRZB1 位的数据传输信号，并将 NRZB1 位的值传送至 NRZ1 位。

<8>　在中断服务程序中将下一个值写入 NRZB1 位。该中断服务程序已由 INTTM5H1 中断启动或通过轮询中断请求标志检测到时序后启动。将下一次要计数的值写入 CR51 中。

<9>　当 NRZ1 位是高电平时，将从 TOH1 引脚输出载波时钟。

<10> 通过执行上述过程，可以获得任意形式的载波时钟。若要停止计数操作，可以将 TMHE1 清零。

如果 CMP01 的值为 N、CMP11 的值为 M、计数时钟频率为 $f_{CNT}$，则载波时钟输出周期与占空比如下所示。

- 载波时钟输出周期 = $(N + M + 2)/f_{CNT}$
- 占空比=高电平宽度/ 载波时钟输出宽度 = （ M + 1） / （N + M + 2）

注意事项 **1.** 在定时器计数操作停止后（**TMHE1=0**），再启动定时器操作（**TMHE1=1**）时，必须设置 **CMP11**（即使是相同的值，也必须对 **CMP11** 重新设置）

**2.** 设置 **TMH1** 的计数时钟频率至少是 **TM51** 的计数时钟频率的 **6** 倍。

**3.** **CMP01** 和 **CMP11** 的值必须在 **01H** 和 **FFH** 之间。

**4.** 定时器计数期间可以改变 **CMP11** 寄存器的值。但是，从改变 **CMP11** 的值到将该值传输到寄存器需要三个操作时钟（由 **TMHMD1** 寄存器的 **CKS12**～**CKS10** 位选择的时钟信号）

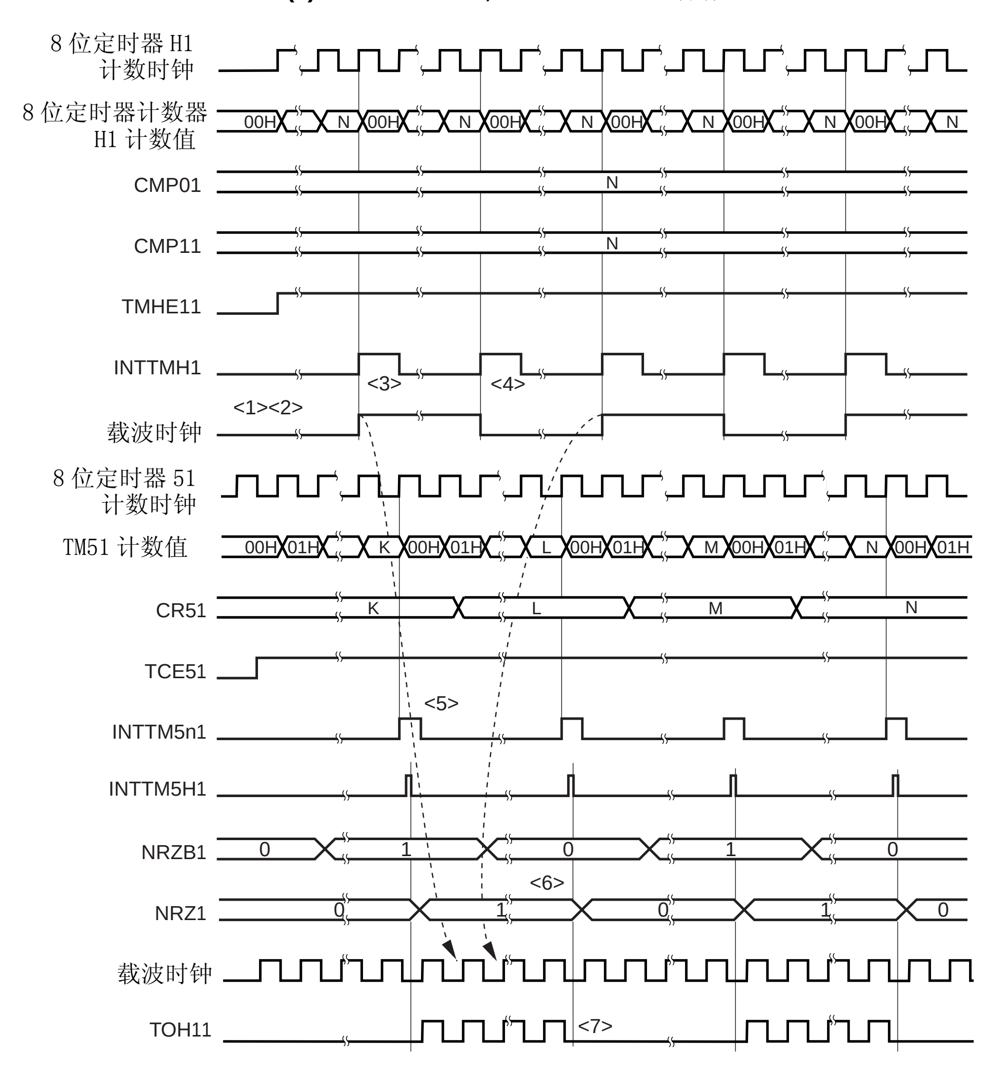
**5.** 在计时操作开始前必须设置 **RMC1**。

备注 **1.** 如需了解输出引脚的设置，参见 **8.3 (3) 端口模式寄存器 1（PM1）**

**2.** INTTMH1 中断信号的使用参见 **第十八章 中断功能**。

图 8-15. 载波发生器模式操作时序 （1/3）

(a) 当 CMP01 = N, CMP11 = N 时的操作




<1>　当 TMHE1 = 0 且 TCE51 = 0 时，停止 8 位定时器计数器 H1 的操作。

<2>　当 TMHE1 = 1 时，8 位定时器计数器 H1 开始计数。此时载波时钟保持默认电平状态。

<3>　当 8 位定时器计数器 H1 的计数值与 CMP01 的值相等时，产生第 1 个 INTTMH1 信号并反转载波时钟信号，同时与 8 位定时器计数器 H1 比较的比较寄存器从 CMP01 切换为 CMP11。8 位定时器计数器 H1 的值被清零（00H）。

<4>　当 8 位定时器计数器 H1 的计数值与 CMP11 的值相等时，产生 INTTMH1 信号并反转载波时钟信号，同时与 8 位定时器计数器 H1 比较的比较寄存器从 CMP11 切换为 CMP01。8 位定时器计数器 H1 的值被清零（00H）。重复执行过程<3>和<4>，将产生 1 个载波时钟，且占空比恒为 50%。

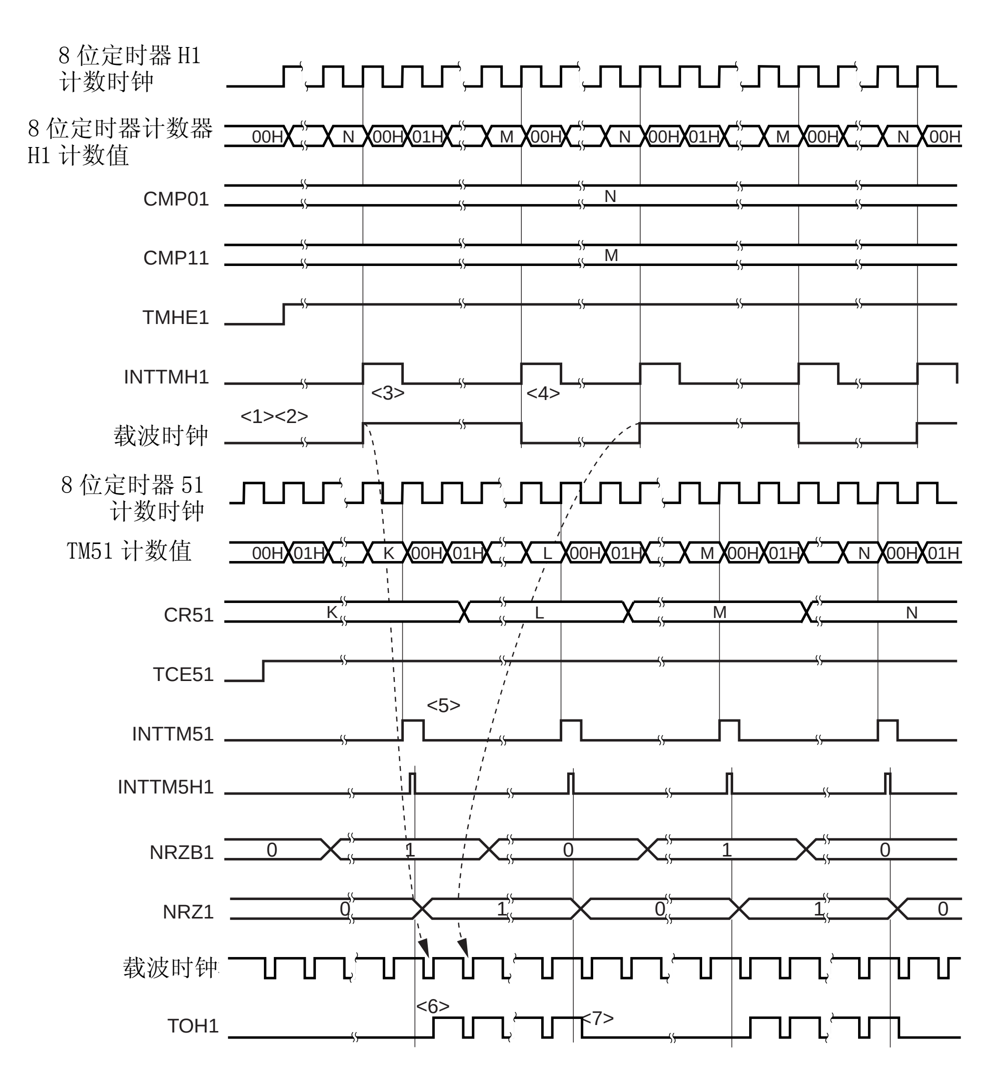
<5>　当 INTTM51 信号产生时，它与 8 位定时器 H1 计数时钟同步，并作为 INTTM5H1 信号输出。

<6>　INTTM5H1 信号用作 NRZB1 位的数据传输信号，并将 NRZB1 位的值传送至 NRZ1 位。

<7>　设置 NRZ1 = 0 时，TOH1 输出低电平。

**图 8-15. 载波发生器模式操作的时序 （2/3）**

**（b） 当 CMP01 = N， CMP11 = M 时的操作**




<1> 当 TMHE1 = 0 且 TCE51 = 0 时，停止 8 位定时器计数器 H1 的操作。

<2> 当 TMHE1 = 1 时，8 位定时器计数器 H1 开始计数。此时载波时钟保持默认电平状态。

<3> 当 8 位定时器计数器 H1 的计数值与 CMP01 的值相等时，产生第 1 个 INTTMH1 信号并反转载波时钟信号，同时与 8 位定时器计数器 H1 比较的比较寄存器从 CMP01 切换为 CMP11。8 位定时器计数器 H1 的值被清零（00H）。

<4> 当 8 位定时器计数器 H1 的计数值与 CMP11 的值相等时，产生 INTTMH1 信号并反转载波时钟信号，同时与 8 位定时器计数器 H1 比较的比较寄存器从 CMP11 切换为 CMP01。8 位定时器计数器 H1 的值被清零（00H）。重复执行过程<3>和<4>，将产生 1 个载波时钟，且占空比不等于 50%。

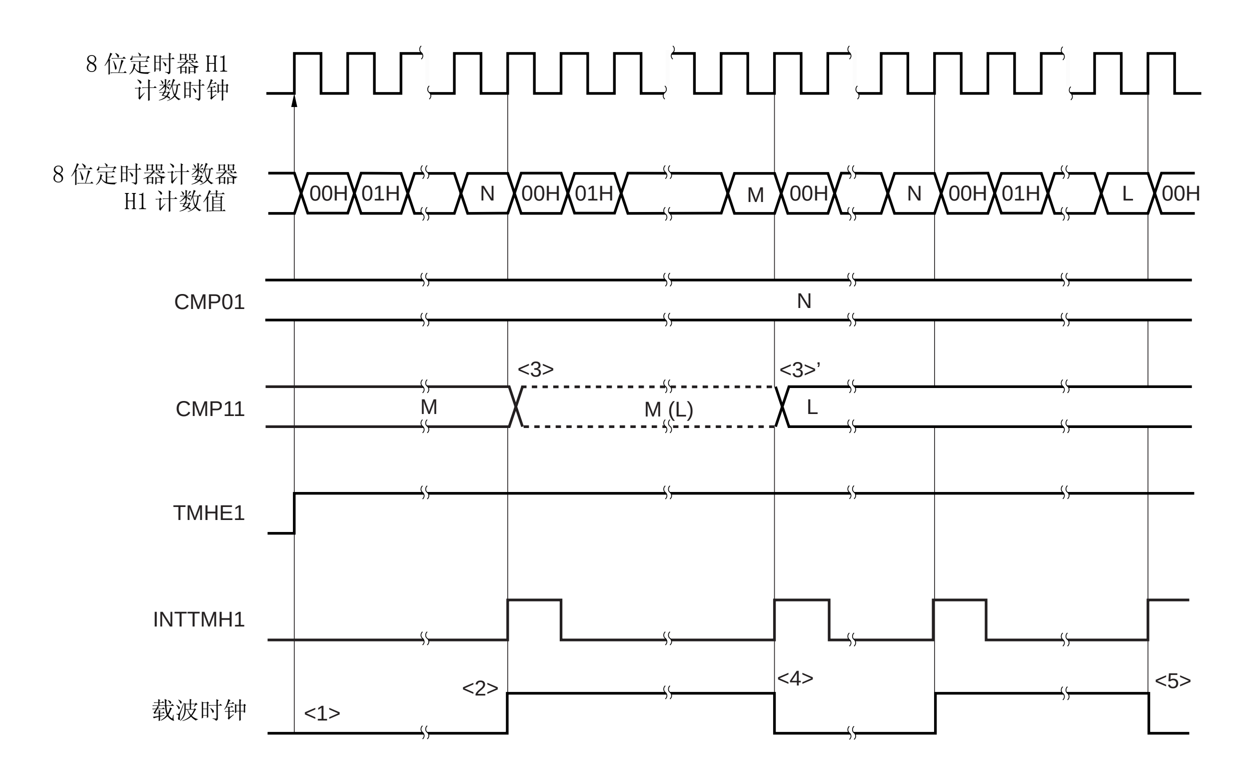
<5> 当 INTTM51 信号产生时，它与 8 位定时器 H1 计数时钟同步，并作为 INTTM5H1 信号输出。

<6> 若 NRZ1=1，在载波时钟的第 1 个上升沿处输出 1 个载波信号。

<7> 当 NRZ1 = 0 时，TOH1 输出保持高电平状态，在载波时钟为高电平时（在过程<6>和<7>中，保证载波时钟波形的高电平宽度）不会转变为低电平。

图 8-15. 载波发生器模式操作的时序 （3/3）

（c） 当 CMP11 改变时的操作




<1> 当 TMHE1 = 1 时，8 位定时器 H1 开始计数。此时载波时钟保持默认电平状态。

<2> 当 8 位定时器计数器 H1 的计数值与 CMP01 的值相等时，输出 INTTMH1 信号、反转载波信号，并将定时器计数器清零(00H)。同时与 8 位定时器计数器 H1 比较的比较寄存器从 CMP01 切换为 CMP11。

<3> CMP11 与计数时钟不同步，在 8 位定时器 H1 操作期间可以修改 CMP11 的值，但修改后的新值（L）被锁存。当 8 位定时器计数器 H1 的计数值与 CMP11 修改前的值（M）相等（<3>'）时，修改 CMP11 的值(<3>')。

但从改变 CMP11 的值到将该值传送到寄存器需要 3 个计数时钟。即使在 3 个计数时钟内产生相等信号，新值不会被传送到寄存器

<4> 当 8 位定时器计数器 H1 的计数值与 CMP11 修改前的值（M）相等时，输出 INTTMH1 信号、反转载波信号，并将定时器计数器清零(00H)。同时与 8 位定时器计数器 H1 比较的比较寄存器从 CMP11 切换为 CMP01。

<5> 由修改后的值（L）确定 8 位定时器计数器 H1 的计数值与 CMP11 的值再次相等时的时序。

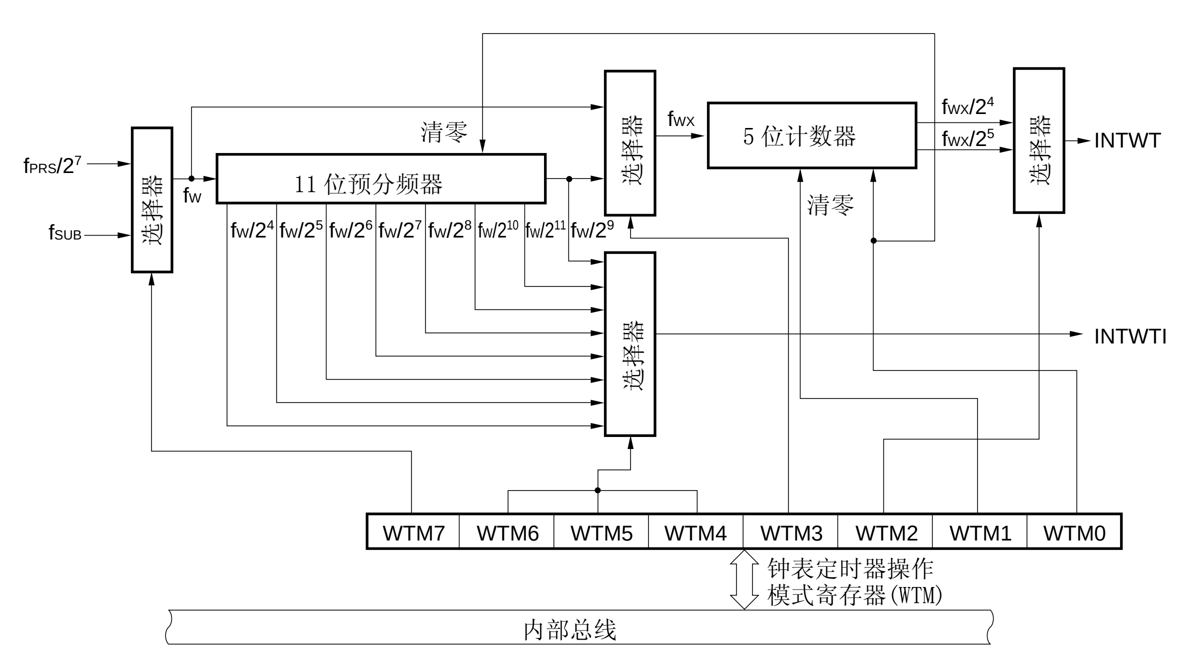
## 9.1 钟表定时器的功能

钟表定时器有以下功能。

- 钟表定时器
- 间隔定时器

钟表定时器和间隔定时器可以同时使用。
图 9-1 为钟表定时器框图。

**图 9-1. 钟表定时器框图**




**备注**　$f_{PRS}$: 外部硬件时钟频率
$f_{SUB}$:副系统时钟频率
$f_W$：钟表定时器时钟频率 ($f_{PRS}/2^7$ 或 $f_{SUB}$)
$f_{WX}$：$f_W$ 或 $f_W/2^9$

**(1)　钟表定时器**

当使用外部硬件时钟或副系统时钟时，以预设的时间间隔产生中断请求信号(INTWT)。

**表 9-1.　钟表定时器中断时间**

| 中断时间 | 操作频率 $f_{SUB}$ = 32.768 kHz | 操作频率 $f_{PRS}$ = 2 MHz | 操作频率 $f_{PRS}$ = 5 MHz | 操作频率 $f_{PRS}$ = 10 MHz | 操作频率 $f_{PRS}$ = 20 MHz |
|---|---|---|---|---|---|
| $2^4/f_W$ | 488 μs | 1.02 ms | 410 μs | 205 μs | 102 μs |
| $2^5/f_W$ | 977 μs | 2.05 ms | 819 μs | 410 μs | 205 μs |
| $2^{13}/f_W$ | 0.25 s | 0.52 s | 0.210 s | 0.105 s | 52.5 ms |
| $2^{14}/f_W$ | 0.5 s | 1.05 s | 0.419 s | 0.210 s | 0.105 s |

备注　$f_{PRS}$:　外部硬件时钟频率

$f_{SUB}$:　副系统时钟频率

$f_W$：　钟表定时器时钟频率($f_{PRS}/2^7$ 或 $f_{SUB}$)

**(2) 间隔定时器**

以预设的时间间隔产生中断请求信号(INTWTI)。

**表 9-2.　间隔定时器间隔时间**

| 间隔时间 | 操作频率 $f_{SUB}$ = 32.768 kHz | 操作频率 $f_{PRS}$ = 2 MHz | 操作频率 $f_{PRS}$ = 5 MHz | 操作频率 $f_{PRS}$ = 10 MHz | 操作频率 $f_{PRS}$ = 20 MHz |
|---|---|---|---|---|---|
| $2^4/f_W$ | 488 μs | 1.02 ms | 410 μs | 205 μs | 102 μs |
| $2^5/f_W$ | 977 μs | 2.05 ms | 820 μs | 410 μs | 205 μs |
| $2^6/f_W$ | 1.95 ms | 4.10 ms | 1.64 ms | 820 μs | 410 μs |
| $2^7/f_W$ | 3.91 ms | 8.20 ms | 3.28 ms | 1.64 ms | 820 μs |
| $2^8/f_W$ | 7.81 ms | 16.4 ms | 6.55 ms | 3.28 ms | 1.64 ms |
| $2^9/f_W$ | 15.6 ms | 32.8 ms | 13.1 ms | 6.55 ms | 3.28 ms |
| $2^{10}/f_W$ | 31.3 ms | 65.5 ms | 26.2 ms | 13.1 ms | 6.55 ms |
| $2^{11}/f_W$ | 62.5 ms | 131.1 ms | 52.4 ms | 26.2 ms | 13.1 ms |

备注　$f_{PRS}$: 外部硬件时钟频率

$f_{SUB}$: 副系统时钟频率

$f_W$:　钟表定时器时钟频率 ($f_{PRS}/2^7$ 或 $f_{SUB}$)

## 9.2　钟表定时器的配置

钟表定时器包括以下硬件。

**表 9-3.　钟表定时器的配置**

| 项目 | 配置 |
|---|---|
| 计数器 | 5 位 × 1 |
| 预分频器 | 11 位 × 1 |
| 控制寄存器 | 钟表定时器操作模式寄存器 (WTM) |

## 9.3 控制钟表定时器的寄存器

由钟表定时器操作模式寄存器(WTM)控制钟表定时器。

- **钟表定时器操作模式寄存器(WTM)**
  该寄存器用于设置钟表定时器计数时钟、允许/禁止操作、预分频器间隔时间和 5 位计数器操作控制。
  可以由 1 位或 8 位存储器操作指令设置 WTM。
  复位信号的产生将 WTM 清零(00H)。

**图 9-2. 钟表定时器操作模式寄存器 (WTM)的格式**

地址： FF6FH　复位后： 00H　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| WTM | WTM7 | WTM6 | WTM5 | WTM4 | WTM3 | WTM2 | WTM1 | WTM0 |

| WTM7 | 钟表定时器计数时钟选择($f_W$) | | | | | |
|---|---|---|---|---|---|---|
| | | $f_{SUB}$ = 32.768 kHz | $f_{PRS}$ = 2 MHz | $f_{PRS}$ = 5 MHz | $f_{PRS}$ = 10 MHz | $f_{PRS}$ = 20 MHz |
| 0 | $f_{PRS}/2^7$ | – | 15.625 kHz | 39.062 kHz | 78.125 kHz | 156.25 kHz |
| 1 | $f_{SUB}$ | 32.768 kHz | – | | | |

| WTM6 | WTM5 | WTM4 | 预分频器间隔时间选择 |
|---|---|---|---|
| 0 | 0 | 0 | $2^4/f_W$ |
| 0 | 0 | 1 | $2^5/f_W$ |
| 0 | 1 | 0 | $2^6/f_W$ |
| 0 | 1 | 1 | $2^7/f_W$ |
| 1 | 0 | 0 | $2^8/f_W$ |
| 1 | 0 | 1 | $2^9/f_W$ |
| 1 | 1 | 0 | $2^{10}/f_W$ |
| 1 | 1 | 1 | $2^{11}/f_W$ |

| WTM3 | WTM2 | 钟表定时器中断时间选择 |
|---|---|---|
| 0 | 0 | $2^{14}/f_W$ |
| 0 | 1 | $2^{13}/f_W$ |
| 1 | 0 | $2^5/f_W$ |
| 1 | 1 | $2^4/f_W$ |

| WTM1 | 5 位计数器操作控制 |
|---|---|
| 0 | 操作停止后清零 |
| 1 | 启动操作 |

| WTM0 | 钟表定时器操作允许 |
|---|---|
| 0 | 停止操作(预分频器和 5 位计数器清零) |
| 1 | 允许操作 |

**注意事项 在钟表定时器操作期间不要修改计数时钟和间隔时间 (通过设置 WTM 的第 4 ～ 7 位 (WTM4 ～ WTM7)实现)。**

备注　**1.** $f_W$： 钟表定时器时钟频率($f_{PRS}$ /$2^7$ 或 $f_{SUB}$)
**2.** $f_{PRS}$：外部硬件时钟频率
**3.** $f_{SUB}$：副系统时钟频率

## 9.4 钟表定时器操作

### 9.4.1 钟表定时器操作

通过使用外部硬件时钟或副系统时钟，钟表定时器以指定的时间间隔产生中断请求信号(INTWT)。

当钟表定时器操作模式寄存器(WTM)的第 0 位(WTM0)和第 1 位(WTM1)为 1 时，开始计数操作。当这些位被设置为零时，将 5 位计数器清零，同时停止计数操作。

在间隔定时器操作的同时，通过将WTM1 清零，钟表定时器的零秒启动被激活。但在这种情况中，不对 11 位预分频器清零。因此，在零秒启动后的第 1 次溢出(INTWT)时会产生 1 个 $2^9 \times 1/f_W$秒的误差。

按以下时间间隔产生中断请求。

**表 9-4. 钟表定时器中断时间**

| WTM3 | WTM2 | 中断时间选择 | 操作频率 $f_{SUB}$ = 32.768 kHz (WTM7 = 1) | 操作频率 $f_{PRS}$ = 2 MHz (WTM7 = 0) | 操作频率 $f_{PRS}$ = 5 MHz (WTM7 = 0) | 操作频率 $f_{PRS}$ = 10 MHz (WTM7 = 0) | 操作频率 $f_{PRS}$ = 20 MHz (WTM7 = 0) |
|---|---|---|---|---|---|---|---|
| 0 | 0 | $2^{14}/f_W$ | 0.5 s | 1.05 s | 0.419 s | 0.210 s | 0.105 s |
| 0 | 1 | $2^{13}/f_W$ | 0.25 s | 0.52 s | 0.210 s | 0.105 s | 52.5 ms |
| 1 | 0 | $2^5/f_W$ | 977 μs | 2.05 ms | 819 μs | 410 μs | 205 μs |
| 1 | 1 | $2^4/f_W$ | 488 μs | 1.02 ms | 410 μs | 205 μs | 102 μs |

备注 **1.** $f_W$: 钟表定时器时钟频率($f_{PRS}/2^7$ 或 $f_{SUB}$)
**2.** $f_{PRS}$: 外部硬件时钟频率
**3.** $f_{SUB}$: 副系统时钟频率

### 9.4.2 间隔定时器操作

钟表定时器用作间隔定时器时，以预设的时间间隔重复产生中断请求(INTWTI)。

可以由钟表定时器操作模式寄存器(WTM)的第 4～6 位(WTM4 ～ WTM6)选择间隔时间。

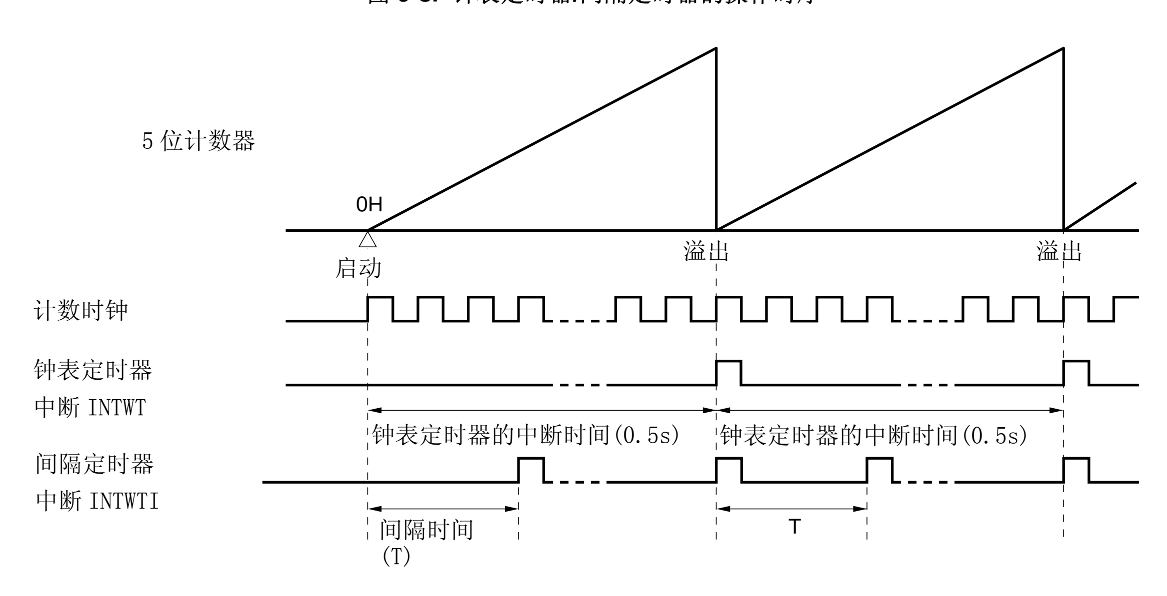
当 WTM 的第 0 位(WTM0)为 1 时，开始计数。而当该位被设置为零时，停止计数。

**表 9-5. 间隔定时器间隔时间**

| WTM6 | WTM5 | WTM4 | 间隔时间 | 操作频率 $f_{SUB}$ = 32.768 kHz (WTM7 = 1) | 操作频率 $f_{PRS}$ = 2 MHz (WTM7 = 0) | 操作频率 $f_{PRS}$ = 5 MHz (WTM7 = 0) | 操作频率 $f_{PRS}$ = 10 MHz (WTM7 = 0) | 操作频率 $f_{PRS}$ = 20 MHz (WTM7 = 0) |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | $2^4/f_W$ | 488 μs | 1.02 ms | 410 μs | 205 μs | 102 μs |
| 0 | 0 | 1 | $2^5/f_W$ | 977 μs | 2.05 ms | 820 μs | 410 μs | 205 μs |
| 0 | 1 | 0 | $2^6/f_W$ | 1.95 ms | 4.10 ms | 1.64 ms | 820 μs | 410 μs |
| 0 | 1 | 1 | $2^7/f_W$ | 3.91 ms | 8.20 ms | 3.28 ms | 1.64 ms | 820 μs |
| 1 | 0 | 0 | $2^8/f_W$ | 7.81 ms | 16.4 ms | 6.55 ms | 3.28 ms | 1.64 ms |
| 1 | 0 | 1 | $2^9/f_W$ | 15.6 ms | 32.8 ms | 13.1 ms | 6.55 ms | 3.28 ms |
| 1 | 1 | 0 | $2^{10}/f_W$ | 31.3 ms | 65.5 ms | 26.2 ms | 13.1 ms | 6.55 ms |
| 1 | 1 | 1 | $2^{11}/f_W$ | 62.5 ms | 131.1 ms | 52.4 ms | 26.2 ms | 13.1 ms |

备注 **1.** $f_W$: 钟表定时器时钟频率($f_{PRS}/2^7$ 或 $f_{SUB}$)
**2.** $f_{PRS}$: 外部硬件时钟频率
**3.** $f_{SUB}$: 副系统时钟频率

图 **9-3.** 钟表定时器/间隔定时器的操作时序



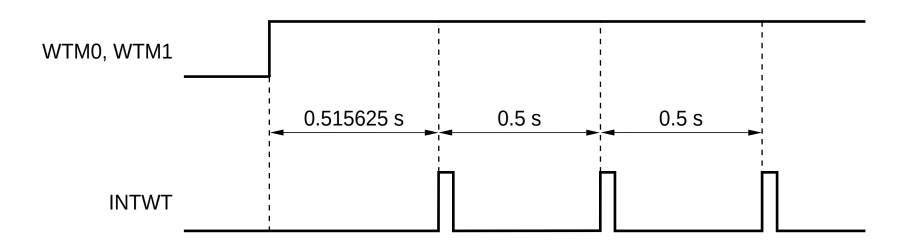

**备注**　fw: 钟表定时器时钟频率
图中括号里的值为 fw = 32.768 kHz (WTM7 = 1, WTM3, WTM2 = 0, 0) 时的操作结果

## 9.5 钟表定时器使用注意事项

通过钟表定时器模式控制寄存器(WTM)(WTM 的第 0 位(WTM0)和第 1 位(WTM1)置 1)允许钟表定时器和 5 位计数器操作时，从寄存器被设置后到第 1 个中断请求(INTWT)产生的这段间隔时间与 WTM 的第 2 和 3 位(WTM2 和 WTM3)指定的间隔时间并不完全相等。但其后的 INTWT 信号会以指定的时间间隔产生。

图 **9-4.** 钟表定时器中断请求信号 **(INTWT)** 产生示例 **(**当中断周期 **= 0.5 s)**

经过 0.515625 秒产生第 1 个INTWT($2^9$ × 1/32768 = 0.015625 s)。
随后 INTWT 每 0.5 秒产生 1 次。



# 第十章　看门狗定时器

## 10.1 看门狗定时器的功能

看门狗定时器使用内部低速振荡时钟。

看门狗定时器用于检测不期望出现的程序循环。如果检测到一个程序循环，将产生一个内部复位信号。

出现以下情况时检测程序循环。

- 如果看门狗定时器计数器溢出
- 如果对看门狗定时器使能寄存器(WDTE)执行 1 位操作指令。
- 如果将“ACH”以外的数据写入 WDTE
- 如果在窗口关闭期间将数据写入 WDTE
- 如果从没有通过 IMS 和 IXS 寄存器设置的区域获取指令(在 CPU 挂起时检查是否有无效校验)
- 如果因为执行 1 条读/写指令，CPU 访问了没有通过 IMS 和 IXS 寄存器设置的区域(FB00H ～ FFFFH 除外) (CPU 程序循环期间检测是否有非法访问)

当由看门狗定时器产生复位时，复位控制标志寄存器（RESF）的第 4 位（WDTRF）置 1。需要了解 RESF 的详细信息，可以参阅 **第二十一章 复位功能**。

## 10.2 看门狗定时器的配置

看门狗定时器包含以下硬件。

**表 10-1. 看门狗定时器的配置**

| 项目 | 配置 |
|---|---|
| 控制寄存器 | 看门狗定时器使能寄存器 (WDTE) |

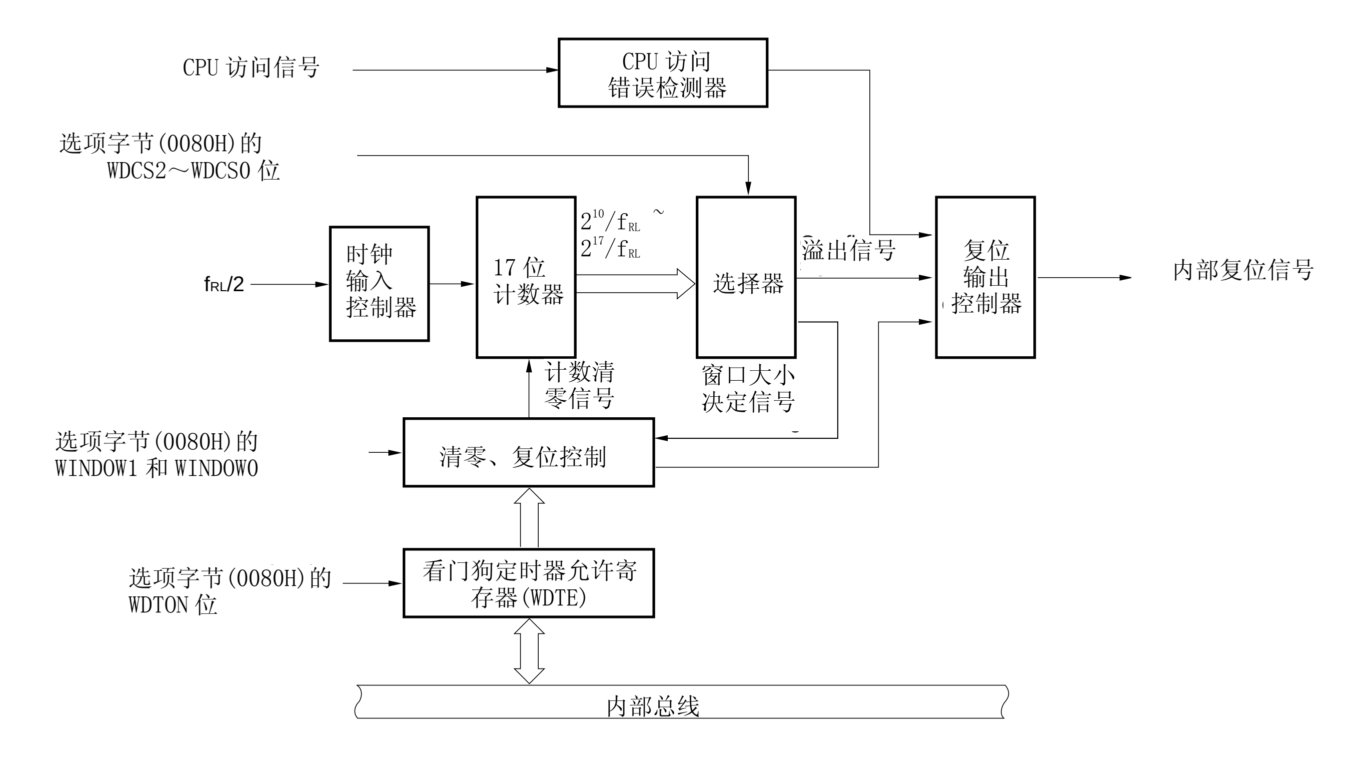
由选项字节设置如何控制计数器的操作、溢出时间和窗口打开周期。

**表 10-2. 选项字节和看门狗定时器的设置**

| 看门狗定时器的设置 | 选项字节 (0080H) |
|---|---|
| 窗口打开周期 | 第6位和第5位(WINDOW1, WINDOW0) |
| 控制看门狗定时器的计数器操作 | 第4位 (WDTON) |
| 看门狗定时器的溢出时间 | 第3位 ～ 第1位(WDCS2 ～ WDCS0) |

**备注**　　需要了解选项字节的内容，可参见 **第二十四章 选项字节**。

**图 10-1. 看门狗定时器的框图**



## 10.3 控制看门狗定时器的寄存器

看门狗定时器由看门狗定时器允许寄存器(WDTE)控制。

**(1) 看门狗定时器允许寄存器(WDTE)**

将 ACH 写入 WDTE 可以对看门狗定时器计数器清零并再次开始计数操作。

可以由 8 位存储器操作指令设置该寄存器。

复位信号的产生将该寄存器设置为 9AH或 1AH[注]。

图 **10-2.** 看门狗定时器允许寄存器 **(WDTE)**的格式

FF99H　　复位 9AH/1AH[注]　　R/W

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| WDTE | | | | | | | | |

注　WDTE 的复位值随选项字节(0080H)的 WDTON 位的设置值而变化。 如果要对看门狗定时器进行操作，设置 WDTON = 1。

| WDTON设置值 | WDTE复位值 |
|---|---|
| 0 (禁止看门狗定时器计数的操作) | 1AH |
| 1 (允许看门狗定时器计数的操作) | 9AH |

注意事项 **1.** 如果写入 **WDTE** 的值不是 **ACH**，将产生一个内部复位信号。如果看门狗定时器的时钟源被停止，则当看门狗定时器的时钟源恢复操作时，将产生内部复位信号。

**2.** 如果使用 **1** 位存储器操作指令对 **WDTE** 进行操作，将产生一个内部复位信号。如果看门狗定时器的时钟源被停止，当提供给看门狗定时器的时钟源恢复操作时，将产生内部复位信号。

**3.** 从 **WDTE** 读取的值为 **9AH/1AH**（与写入值（**ACH**）不同）。

## 10.4 看门狗定时器的操作

### 10.4.1 看门狗定时器操作的控制

1. 当使用看门狗定时器时，由选项字节(0080H)指定其操作。

    - 通过设置选项字节(0080H)的第 4 位(WDTON)=1，允许看门狗定时器的计数操作(复位释放后计数器开始计数)(如需了解详细信息，可参见 **第二十六章**)。

| WDTON | 看门狗定时器计数器的操作控制/非法访问检测 |
|---|---|
| 0 | 禁止计数器操作 (复位后停止计数)，禁止非法访问检测操作 |
| 1 | 允许计数器操作 (复位后开始计数)，允许进行非法访问检测操作 |

    - 通过使用选项字节(0080H)的第 3 位～第 1 位(WDCS2 ～ WDCS0)设置溢出时间(如需了解详细信息，参见 **10.4.2** 和 **第二十六章**)。
    - 通过使用选项字节(0080H)的第 6 位和第 5 位(WINDOW1 和 WINDOW0)设置窗口打开周期(如需了解详细信息，参见 **10.4.3** 和 **第二十六章**)。

2. 复位释放后，看门狗定时器开始计数。
3. 在看门狗定时器开始计数后且在达到选项字节设置的溢出时间之前，将"ACH"写入 WDTE，看门狗定时器将被清零并再次开始计数。
4. 经过上述步骤后对 WDTE 进行第二次写操作，或在复位释放后的窗口打开期间对 WDTE 进行第二次写操作。如果在窗口关闭期间对 WDTE 进行写操作，则会产生内部复位信号。
5. 如果已经历了溢出时间还没有将"ACH"写入 WDTE，则会产生内部复位信号。
    出现以下情况时会产生内部复位信号。

    - 如果对看门狗定时器允许寄存器(WDTE)执行位操作指令时。
    - 如果将"ACH"以外的数据写入 WDTE。
    - 如果从没有通过 IMS 和 IXS 寄存器设置的区域获取指令(在 CPU 程序循环期间检查是否有无效校验)
    - 如果因为执行 1 条读/写指令，CPU 访问了没有通过 IMS 和 IXS 寄存器设置的区域(FB00H ～ FFFFH 除外)(CPU 程序循环期间检测是否有非法访问)

注意事项　**1.** **复位释放后对 WDTE 进行第一次写操作，将清零看门狗定时器，如果这一操作是在到达溢出时间之前进行的而不管写操作时序，看门狗定时器再次开始计数。**

**2.** **如果通过将"ACH"写入 WDTE 来对看门狗定时器清零，则实际的溢出时间可能与由选项字节设置的溢出时间不同，最高可达 2/$f_{RL}$ 秒。**

**3.** **在计数值溢出(FFFFH)之前，可将看门狗定时器立即清零。**

注意事项 **4.** 在 **HALT** 和 **STOP** 模式下看门狗定时器的操作依据选项字节第 **0** 位**(LSROSC)**的设置值的不同而不同，如下所示。

| | LSROSC = 0 (可由软件停止内部低速振荡器) | LSROSC = 1 (不能停止内部低速振荡器) |
|---|---|---|
| HALT 模式 | 看门狗定时器操作停止 | 看门狗定时器操作继续 |
| STOP 模式 | | |

如果 **LSROSC = 0**，则在 **HALT** 或 **STOP** 模式释放后看门狗定时器恢复计数。此时，不对计数器清零，而从计数器停止时的值开始计数。

如果在 **LSROSC = 0** 时通过设置 **LSRSTOP(**内部振荡模式寄存器**(RCM)**的第 **1** 位**=1)**停止内部低速振荡器的振荡，则看门狗定时器停止操作。此时不对计数器清零**(0)**。

**5.** 在闪存自编程和**EEPROM**[TM]仿真期间看门狗定时器不停止操作。在处理期间中断响应被延迟。设置溢出时间和窗口大小时，应考虑延迟。

### 11.4.2 看门狗定时器溢出时间的设置

通过使用选项字节(0080H)的第 3 位～第 1 位(WDCS2 ～ WDCS0)，设置看门狗定时器的溢出时间。

如果出现溢出，则将产生内部复位信号。在到达溢出时间之前且在窗口打开周期内通过将“ACH”写入 WDTE，可对当前计数值清零，且看门狗定时器再次开始计数。

溢出时间设置如下所示。

**表 10-3.** 看门狗定时器溢出时间的设置

| WDCS2 | WDCS1 | WDCS0 | 看门狗定时器的溢出时间 |
|---|---|---|---|
| 0 | 0 | 0 | $2^{10}/f_{RL}$ (3.88 ms) |
| 0 | 0 | 1 | $2^{11}/f_{RL}$ (7.76 ms) |
| 0 | 1 | 0 | $2^{12}/f_{RL}$ (15.52 ms) |
| 0 | 1 | 1 | $2^{13}/f_{RL}$ (31.03 ms) |
| 1 | 0 | 0 | $2^{14}/f_{RL}$ (62.06 ms) |
| 1 | 0 | 1 | $2^{15}/f_{RL}$ (124.12 ms) |
| 1 | 1 | 0 | $2^{16}/f_{RL}$ (248.24 ms) |
| 1 | 1 | 1 | $2^{17}/f_{RL}$ (496.48 ms) |

注意事项 **1.** 禁止出现 **WDCS2 = WDCS1 = WDCS0 = 0** 且 **WINDOW1 = WINDOW0 = 0** 的情况。

**2.** 在闪存自编程和 **EEPROM** 仿真期间看门狗定时器不停止操作。在处理期间中断响应被延迟。设置溢出时间和窗口大小时，应考虑延迟。

备注 **1.** $f_{RL}$: 内部低速振荡时钟频率

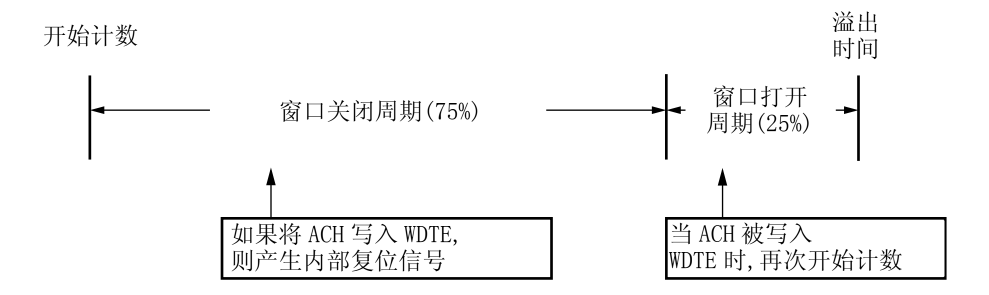
**2.** ( ): $f_{RL}$ = 264 kHz (MAX.)

### 10.4.3 看门狗定时器窗口打开周期的设置

通过使用选项字节(0080H)的第 6 位和第 5 位(WINDOW1, WINDOW0)，设置看门狗定时器的窗口打开周期。窗口使用情况描述如下。

- 如果在窗口打开期间将“ACH”写入 WDTE，则对看门狗定时器清零并再次开始计数。
- 在窗口关闭期间即使将“ACH”写入 WDTE，也会检测到异常情况，并产生内部复位信号。

举例: 如果窗口打开周期为 25%




**注意事项　复位释放后对 WDTE 进行第一次写操作，将清零看门狗定时器，如果这一操作是在到达溢出时间之前进行的而不管写操作时序，看门狗定时器再次开始计数。**

窗口打开周期设置如下所示。

表 **10-4.** 看门狗定时器窗口打开周期的设置

| WINDOW1 | WINDOW0 | 看门狗定时器的窗口打开周期 |
|---|---|---|
| 0 | 0 | 25% |
| 0 | 1 | 50% |
| 1 | 0 | 75% |
| 1 | 1 | 100% |

**注意事项　1.　禁止出现 WDCS2 = WDCS1 = WDCS0 = 0 且 WINDOW1 = WINDOW0 = 0 的情况。**

**2.　在闪存自编程和 EEPROM 仿真期间看门狗定时器不停止操作。在处理阶段中断响应被延迟。设置溢出时间和窗口大小时，应考虑延迟。**

**备注**　如果将溢出时间设置为 $2^{10}/f_{RL}$，则窗口关闭时间和打开时间如下所示。

(当 2.7 V ≤ $V_{DD}$ ≤ 5.5 V)

| | 窗口打开周期的设置 | | | |
|---|---|---|---|---|
| | 25% | 50% | 75% | 100% |
| 窗口关闭时间 | 0 ～ 3.56 ms | 0 ～ 2.37 ms | 0 ～ 0.119 ms | 无 |
| 窗口打开时间 | 3.56 ～ 3.88 ms | 2.37 ～ 3.88 ms | 0.119 ～ 3.88 ms | 0 ～ 3.88 ms |

<当窗口打开周期为 25%>

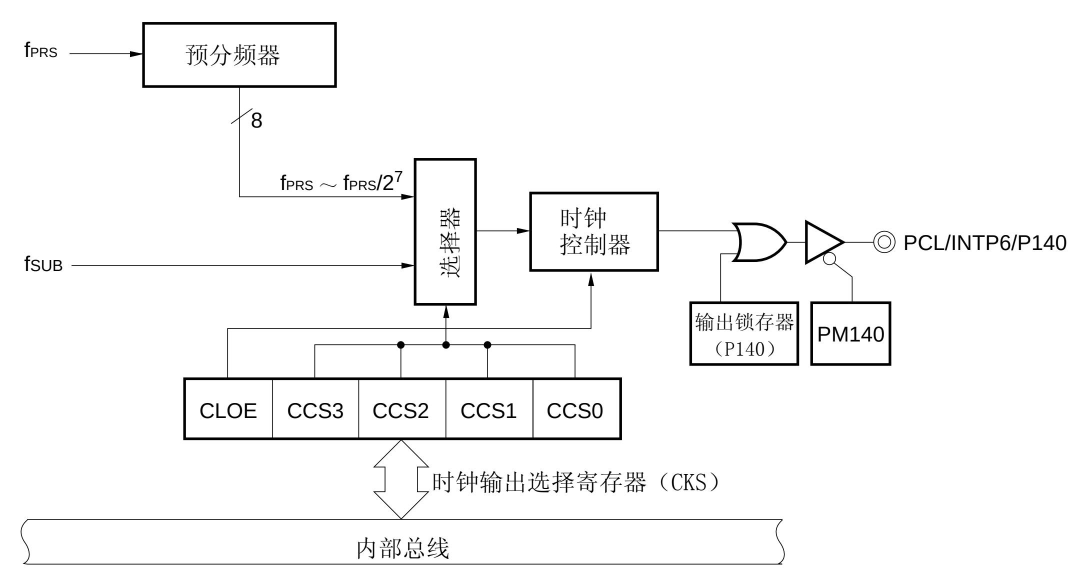
- 溢出时间:
  $2^{10}/f_{RL}$ (MAX.) = $2^{10}$/264 kHz (MAX.) = 3.88 ms
- 窗口关闭时间:
  0 ～ $2^{10}/f_{RL}$ (MIN.) × (1 – 0.25) = 0 ～ $2^{10}$/216 kHz (MIN.) × 0.75 = 0 ～ 3.56 ms
- 窗口打开时间:
  $2^{10}/f_{RL}$ (MIN.) × (1 – 0.25) ～ $2^{10}/f_{RL}$ (MAX.) = $2^{10}$/216 kHz (MIN.) × 0.75 ～ $2^{10}$/264 kHz (MAX.)
  = 3.56 ～ 3.88 ms

## 11.1 时钟输出控制器的功能

时钟输出控制器用于遥控发射期间的载波输出，以及用于外部 IC 的时钟输出。可以输出由时钟输出选择寄存器（CKS）选择的时钟。

图 11-1 显示了时钟输出控制器的框图。

**图 11-1. 时钟输出控制器的框图**



## 11.2 时钟输出控制器的配置

时钟输出控制器包括以下硬件。

**表 11-1. 时钟输出控制器的配置**

| 项目 | 配置 |
| --- | --- |
| 控制寄存器 | 时钟输出选择寄存器 (CKS)<br>端口模式寄存器 14 (PM14)<br>端口寄存器 14 (P14) |

## 11.3 控制时钟输出控制器的寄存器

以下两个寄存器用于控制时钟输出控制器。
- 时钟输出选择寄存器（CKS）
- 端口模式寄存器 14 （PM14）

### (1) 时钟输出选择寄存器 (CKS)

该寄存器用于设置时钟输出（PCL）的输出允许/禁止，并设置输出时钟。
可以由 1 位或 8 位存储器操作指令设置 CKS。
复位信号的产生将 CKS 清零（00H）。

### 图 11-2. 时钟输出选择寄存器（CKS）的格式

地址:  FF40H　复位后:  00H　R/W

| 符号 | 7 | 6 | 5 | <4> | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CKS | 0 | 0 | 0 | CLOE | CCS3 | CCS2 | CCS1 | CCS0 |

| CLOE | PCL 输出允许/禁止 |
|---|---|
| 0 | 停止时钟分频电路的操作。PCL 恒为低电平。 |
| 1 | 允许时钟分频电路的操作。PCL 输出允许。 |

| CCS3 | CCS2 | CCS1 | CCS0 | PCL 输出时钟选择 | | | |
|---|---|---|---|---|---|---|---|
| | | | | | $f_{SUB}$ = 32.768 kHz | $f_{PRS}$ = 10 MHz | $f_{PRS}$ = 20 MHz |
| 0 | 0 | 0 | 0 | $F_{PRS}$[注1] | – | 10 MHz | 禁止设置[注2] |
| 0 | 0 | 0 | 1 | $f_{PRS}/2$ | | 5 MHz | 10 MHz |
| 0 | 0 | 1 | 0 | $f_{PRS}/2^2$ | | 2.5 MHz | 5 MHz |
| 0 | 0 | 1 | 1 | $f_{PRS}/2^3$ | | 1.25 MHz | 2.5 MHz |
| 0 | 1 | 0 | 0 | $f_{PRS}/2^4$ | | 625 kHz | 1.25 MHz |
| 0 | 1 | 0 | 1 | $f_{PRS}/2^5$ | | 312.5 kHz | 625 kHz |
| 0 | 1 | 1 | 0 | $f_{PRS}/2^6$ | | 156.25 kHz | 312.5 kHz |
| 0 | 1 | 1 | 1 | $f_{PRS}/2^7$ | | 78.125 kHz | 156.25 kHz |
| 1 | 0 | 0 | 0 | $f_{SUB}$ | 32.768 kHz | – | |
| 其它 | | | | 禁止设置 | | | |

注　**1.** 当 1.8 V ≤ $V_{DD}$ < 2.7 V 时，如果外围硬件时钟使用内部高速振荡时钟操作时，禁止设置 CCS3 = CCS2 = CCS1 = CCS0 = 0 (PCL 的输出时钟: $f_{PRS}$)。

　　**2.** 禁止设置超过 10 MHz 的 PCL 输出时钟频率。

**注意事项　当时钟输出操作停止时(CLOE = 0)，设置 CCS3 ～ CCS0。**

备注　**1.** $f_{PRS}$:  外围硬件时钟频率

　　　**2.** $f_{SUB}$:  副系统时钟频率

**(2) 端口模式寄存器 14 (PM14)**

该寄存器可位选端口 14 的输入/输出操作模式。

当使用 P140/INTP6/PCL 引脚作为时钟输出时，设置 PM140 和 P140 的输出锁存器的值为 0。

可以由 1 位或 8 位存储器操作指令设置 PM14。

复位信号产生将 PM14 设置为 FFH。

**图 11-3. 端口模式寄存器 14 (PM14) 的格式**

地址: FF2EH 复位后: FFH R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM14 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | PM140 |

| PM140 | P140 引脚 I/O 模式选择 |
|---|---|
| 0 | 输出模式 (打开输出缓冲器) |
| 1 | 输入模式 (关闭输出缓冲器) |

## 11.4 时钟输出控制器的操作

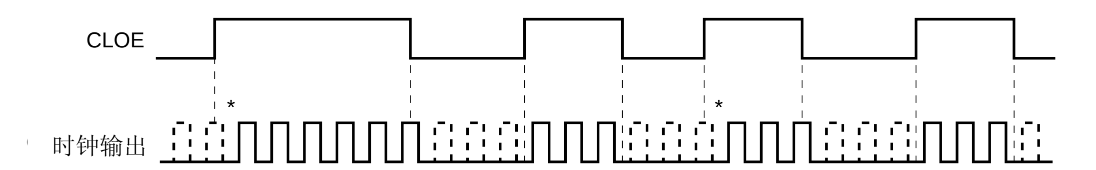
时钟脉冲输出包含以下过程。

<1> 通过时钟输出选择寄存器（CKS）的第 0 位～第 3 位（CCS0 ~ CCS3）选择时钟脉冲输出频率（时钟脉冲输出为禁止状态）。

<2> 将 CKS 的第 4 位（CLOE）置 1，允许时钟输出。

**备注** 在时钟输出的输出允许/禁止切换阶段，时钟输出控制器不能输出小宽度的脉冲。如图 11-4 所示，必须确保在时钟的低电平区（图中标 * 处）启动输出。当停止输出时，也必须在时钟高电平之后停止输出。

**图 11-4. 遥控输出应用举例**

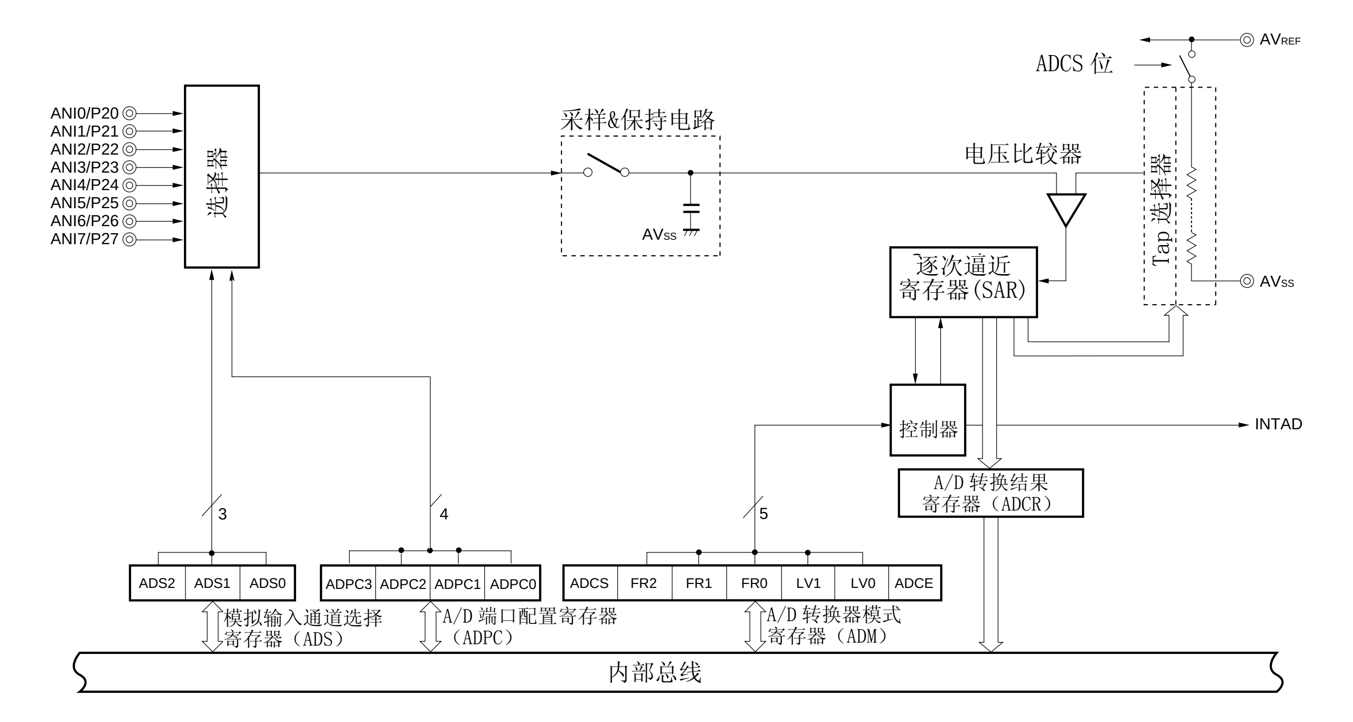


## 12.1 A/D 转换器的功能

A/D 转换器用于将模拟输入信号转换为数字信号，最多可由 8 个通道(ANI0 ～ ANI7)组成，具有 10 位分辨率。
A/D 转换器有以下功能。

- **10 位分辨率 A/D 转换**
  从模拟输入 ANI0～ANI7 中选择一个通道，重复执行 10 位分辨率 A/D 转换。每次 A/D 转换结束，都可以产生一个中断请求(INTAD)。

**图 12-1. A/D 转换器的框图**



## 12.2 A/D 转换器的配置

A/D 转换器包括以下硬件。

**(1) ANI0 ～ ANI7 引脚**

这些是 8 通道 A/D 转换器的模拟输入引脚。用于输入模拟信号，以便转换成数字信号。除了那些被选为模拟输入的引脚外，其它引脚均可用作 I/O 端口引脚。

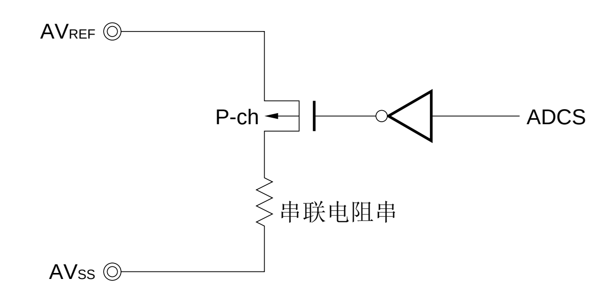
**(2) 采样&保持电路**

采样&保持电路在 A/D 转换器启动时用于采样由选择器选择的模拟输入引脚的输入电压，并在 A/D 转换期间保持采样到的电压值。

**(3) 串联电阻串**

串联电阻串用于连接 $AV_{REF}$ 和 $AV_{SS}$，并产生一个电压，与采样到的电压值进行比较。

**图 12-2. 串联电阻串的电路配置**




**(4) 电压比较器**

电压比较器用于比较采样到的电压值与串联电阻串的输出电压。

**(5) 逐次逼近寄存器 (SAR)**

该寄存器用于转换电压比较器所比较的结果，并从最高有效位(MSB)开始转换结果。

当电压被转换成数字值，并写入最低有效位时(A/D 转换结束)，将 SAR 的内容传送到 A/D 转换结果寄存器(ADCR)中。

**(6) 10 位 A/D 转换结果寄存器 (ADCR)**

每次 A/D 转换结束时，将 A/D 转换结果从逐次逼近寄存器(SAR)传送至该寄存器中。并由该寄存器将转换结果保存在它的高 10 位中(低 6 位恒为 0)。

**(7) 8 位 A/D 转换结果寄存器 (ADCRH)**

每次 A/D 转换结束时，将 A/D 转换结果从逐次逼近寄存器传送至该寄存器中。并将转换结果保存在 ADCRH 的高 8 位中。

注意事项 **当从 ADCR 和 ADCRH 读取数据时，会产生一个等待周期。当 CPU 工作在副系统时钟，且外围硬件时钟处于停止状态时，不要从 ADCR 和 ADCRH 读取数据。需要了解详细信息，可参阅 第三十二章 等待注意事项。**

**(8) 控制器**

该电路用于控制模拟输入信号被转换成数字信号的转换时间，并启动和停止转换操作。当 A/D 转换结束时，该控制器产生中断 INTAD。

**(9) $AV_{REF}$ 引脚**

该引脚为 A/D 转换器输入一个模拟供电/参考电压。当端口 2 被用作数字端口时，该引脚的电压应与 $V_{DD}$ 的电压保持一致。

根据 $AV_{REF}$ 和 $AV_{SS}$ 的电压，将输入到 ANI0 ～ ANI7 的信号转换成数字信号。

**(10) $AV_{SS}$ 引脚**

这是 A/D 转换器的信号地引脚。即便没有使用 A/D 转换器，该引脚的电压也应与 $V_{SS}$ 的电压保持一致。

**(11) A/D 转换器模式寄存器(ADM)**

该寄存器用于设置被转换的模拟输入信号的转换时间，并启动或停止转换操作。

**(12) A/D 端口配置寄存器(ADPC)**

该寄存器用于将 ANI0/P20 ～ANI7/P27 引脚切换成 A/D 转换器的模拟输入或端口的数字 I/O。

**(13) 模拟输入通道选择寄存器 (ADS)**

该寄存器用来选择输入模拟电压(将被转换成数字信号)的端口。

**(14) 端口模式寄存器 2 (PM2)**

该寄存器用于将 ANI0/P20 ～ANI7/P27 引脚切换为输入或输出。

## 12.3 A/D 转换器使用的寄存器

A/D 转换器使用以下六个寄存器。

- A/D 转换器模式寄存器(ADM)
- A/D 端口配置寄存器(ADPC)
- 模拟输入通道选择寄存器(ADS)
- 端口模式寄存器 2(PM2)
- 10 位 A/D 转换结果寄存器(ADCR)
- 8 位 A/D 转换结果寄存器(ADCRH)

### (1) A/D 转换器模式寄存器 (ADM)

该寄存器设置模拟输入的 A/D 转换时间，并启动/停止转换。
可由 1 位或 8 位存储器操作指令设置 ADM。
复位信号产生将该寄存器清零(00H)。

**图 12-3. A/D 转换器模式寄存器 (ADM)的格式**

地址：FF28H　　复位后：00H　R/W

| 符号 | <7> | 6 | 5 | 4 | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| ADM | ADCS | 0 | FR2[注1] | FR1[注1] | FR0[注1] | LV1[注1] | LV0[注1] | ADCE |

| ADCS | A/D 转换操作控制 |
|---|---|
| 0 | 停止转换操作 |
| 1 | 允许转换操作 |

| ADCE | 比较器操作控制[注2] |
|---|---|
| 0 | 停止比较器的操作 |
| 1 | 允许比较器的操作（比较器：1/2AVREF 操作） |

注　**1.** 如需了解 FR2～FR0、LV1、LV0 和 A/D 转换的详细信息，可参见**表 12-2 A/D 转换时间的选择**。

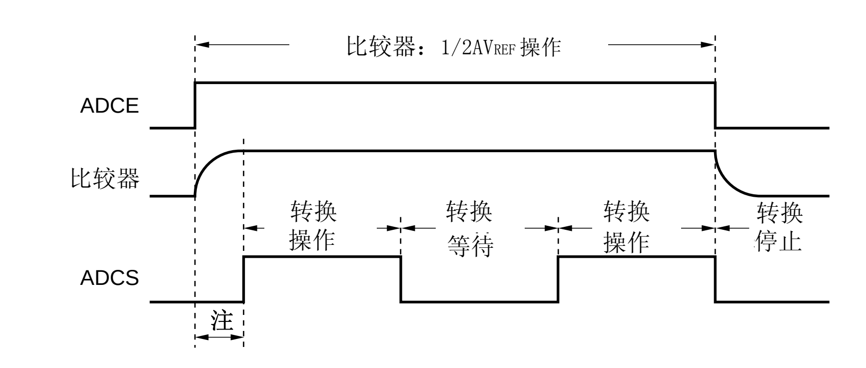
**2.** 比较器的的操作是由 ADCS 和 ADCE 控制的，从操作开始到操作稳定需要 1μs。因此，当 ADCE 被设置为 1 后，至少经过 1μs 的时间，再将 ADCS 置 1 时，此时的转换结果优先与第一次的转换结果。否则，忽略第一次转换的数据。

**表 12-1. ADCS 和 ADCE 的设置**

| ADCS | ADCE | A/D 转换操作 |
|---|---|---|
| 0 | 0 | 停止状态 (不存在直流功耗) |
| 0 | 1 | 转换等待模式 (比较器：1/2AVREF 操作，只有比较器产生功耗) |
| 1 | 0 | 转换模式 (停止比较器的操作[注]) |
| 1 | 1 | 转换模式 （比较器：1/2AVREF 操作） |

注　因为第一次转换的结果不确定，因此忽略第 1 次转换的数据。

图 **12-4.** 使用参考电压发生器时的时序图




注　从 ADCE 位上升到 ADCS 位下降所需时间必须至少为 1 $\mu$s，以便稳定内部电路。

注意事项 **1.** 在将 **FR0** ～ **FR2** 位，**LV1** 和 **LV0** 位修改成其他值之前，**A/D** 转换必须停止。

**2.** 如果对 **ADM** 赋值，会产生 **1** 个等待周期。当 **CPU** 工作在副系统时钟，且外围硬件时钟处于停止状态时，不要对 **ADM** 赋值。需要了解详细信息，可参阅 第三十二章 等待注意事项。

## 表 12-2. A/D 转换时间的选择

### (1) 2.7 V ≤ $AV_{REF}$ ≤ 5.5 V

| A/D 转换器模式寄存器 (ADM) | | | | | 转换时间的选择 | | | | 转换时钟($f_{AD}$) |
|---|---|---|---|---|---|---|---|---|---|
| FR2 | FR1 | FR0 | LV1 | LV0 | | $f_{PRS}$ = 2 MHz | $f_{PRS}$ = 10 MHz | $f_{PRS}$ = 20 MHz[Note] | |
| 0 | 0 | 0 | 0 | 0 | 264/$f_{PRS}$ | 禁止设置 | 26.4 μs | 13.2 μs[注] | $f_{PRS}$/12 |
| 0 | 0 | 1 | 0 | 0 | 176/$f_{PRS}$ | | 17.6 μs | 8.8 μs[注] | $f_{PRS}$/8 |
| 0 | 1 | 0 | 0 | 0 | 132/$f_{PRS}$ | | 13.2 μs | 6.6 μs[注] | $f_{PRS}$/6 |
| 0 | 1 | 1 | 0 | 0 | 88/$f_{PRS}$ | | 8.8 μs[Note] | 禁止设置 | $f_{PRS}$/4 |
| 1 | 0 | 0 | 0 | 0 | 66/$f_{PRS}$ | 33.0 μs | 6.6 μs[Note] | | $f_{PRS}$/3 |
| 1 | 0 | 1 | 0 | 0 | 44/$f_{PRS}$ | 22.0 μs | 禁止设置 | | $f_{PRS}$/2 |
| 其它 | | | | | 禁止设置 | | | | |

注 只有当 4.0 V ≤ $AV_{REF}$ ≤ 5.5 V 时，才能设置。

### (2) 2.3 V ≤ $AV_{REF}$ < 2.7 V

| A/D 转换器模式寄存器 (ADM) | | | | | 转换时间的选择 | | | 转换时钟($f_{AD}$) |
|---|---|---|---|---|---|---|---|---|
| FR2 | FR1 | FR0 | LV1 | LV0 | | $f_{PRS}$ = 2 MHz | $f_{PRS}$ = 5 MHz | |
| 0 | 0 | 0 | 0 | 1 | 480/$f_{PRS}$ | 禁止设置 | 禁止设置 | $f_{PRS}$/12 |
| 0 | 0 | 1 | 0 | 1 | 320/$f_{PRS}$ | | 64.0 μs | $f_{PRS}$/8 |
| 0 | 1 | 0 | 0 | 1 | 240/$f_{PRS}$ | | 48.0 μs | $f_{PRS}$/6 |
| 0 | 1 | 1 | 0 | 1 | 160/$f_{PRS}$ | | 32.0 μs | $f_{PRS}$/4 |
| 1 | 0 | 0 | 0 | 1 | 120/$f_{PRS}$ | 60.0 μs | 禁止设置 | $f_{PRS}$/3 |
| 1 | 0 | 1 | 0 | 1 | 80/$f_{PRS}$ | 40.0 μs | 禁止设置 | $f_{PRS}$/2 |
| 其它 | | | | | 禁止设置 | | | |

注意事项 **1.** 根据以下条件设置转换时间
- **4.0 V ≤ $AV_{REF}$ ≤ 5.5 V: $f_{AD}$ = 0.6 ～ 3.6 MHz**
- **2.7 V ≤ $AV_{REF}$ < 4.0 V: $f_{AD}$ = 0.6 ～ 1.8 MHz**
- **2.3 V ≤ $AV_{REF}$ < 2.7 V: $f_{AD}$ = 0.6 ～ 1.48 MHz**

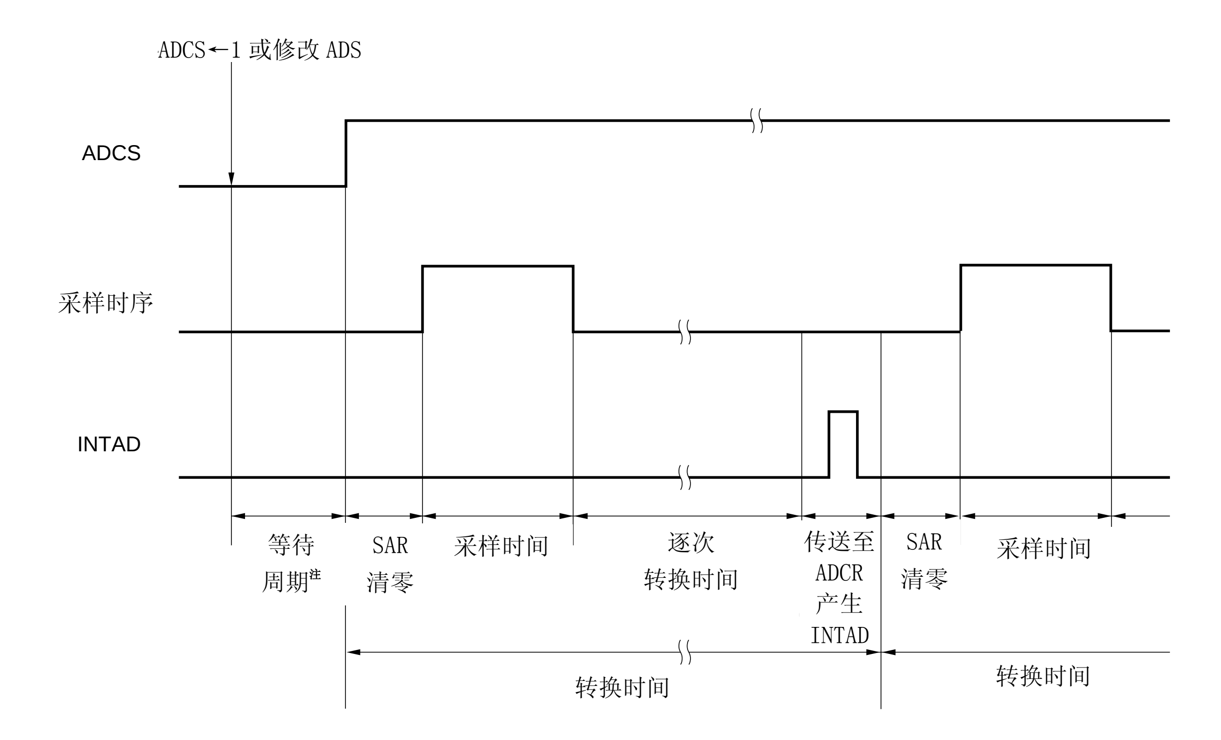
**2.** 当修改 **FR2～FR0**，**LV1** 和 **LV0** 为其他值时，要预先停止 **A/D** 转换一次 **(ADCS = 0)** 。

**3.** 当 **2.3 V ≤ $AV_{REF}$ < 2.7 V** 时，修改 **LV1** 和 **LV0** 的默认值。

**4.** 上述的转换时间不包括时钟频率误差。在选择转换时间的时候，要将时钟频率误差考虑进去。

备注　$f_{PRS}$: 外围硬件时钟频率

图 12-5. A/D 转换器采样和 A/D 转换时序




注　等待周期的详细信息参见 第三十二章 等待注意事项 。

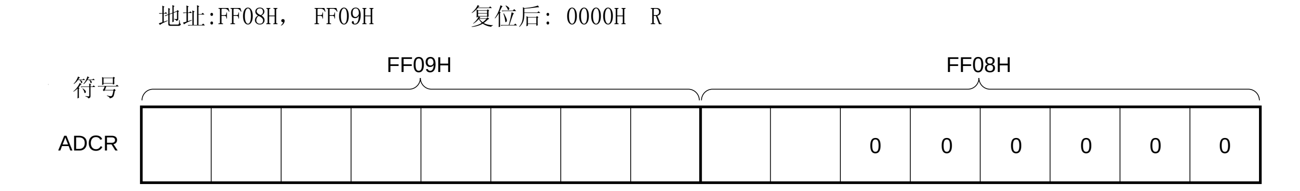
**(2) 10 位 A/D 转换结果寄存器 (ADCR)**

这是一个 16 位寄存器，用来存储 A/D 转换结果。寄存器的低 6 位恒为 0。每次 A/D 转换结束时，将转换结果从逐次逼近寄存器传送至 ADCR 中。转换结果的高 8 位存储在 FF09H 中，而转换结果的低 2 位存储在 FF08H 的高 2 位中。

可由 16 位存储器操作指令读取 ADCR。

复位信号产生将该寄存器清零(0000H)。

图 12-6. 10 位 A/D 转换结果寄存器(ADCR)的格式




注意事项 **1.** 在对 **A/D** 转换器模式寄存器**(ADM)**和模拟输入通道选择寄存器**(ADS)**和 **A/D** 端口配置寄存器**(ADPC)** 赋值时，**ADCR** 的内容可能是不确定的。转换结束后先读取转换结果，再对 **ADM**，**ADS** 和 **ADPC** 赋值。如果没有按照上述时序，可能会读取到不正确的转换结果。

**2.** 如果从 **ADCR** 读取数据，会产生一个等待周期。当 **CPU** 工作在副系统时钟，且外围硬件时钟处于停止状态时，不要从 **ADCR** 读取数据。需要了解详细信息，可参阅 第三十二章 等待注意事项 。

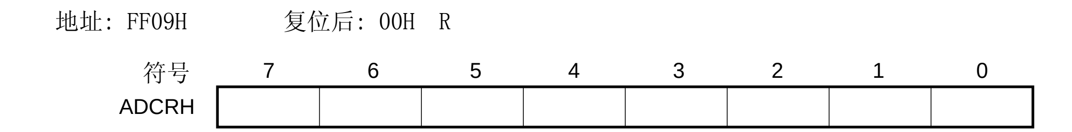
**(3)　8 位 A/D 转换结果寄存器 (ADCRH)**

这是一个 8 位寄存器，用来存储 A/D 转换结果。可以存储 10 位分辨率的高 8 位。

可由 8 位存储器操作指令读取 ADCRH。

复位信号产生将该寄存器清零(00H)。

**图 12-7. 8 位 A/D 转换结果寄存器 (ADCRH) 的格式**




**注意事项　1. 在对 A/D 转换器模式寄存器(ADM)、模拟输入通道选择寄存器(ADS)和 A/D 端口配置寄存器(ADPC) 赋值时，ADCRH 的内容可能不确定。转换结束后先读取转换结果，再对 ADM、ADS 和 ADPC 赋值。如果没有按照上述时序，可能会读取到不正确的转换结果。**

**2. 如果从 ADCRH 读取数据，会产生一个等待周期。当 CPU 工作在副系统时钟，且外围硬件时钟处于停止状态时，不要从 ADCRH 读取数据。需要了解详细信息，可参阅 第三十二章 等待注意事项。**

**(4) 模拟输入通道选择寄存器 (ADS)**

该寄存器用来选择被转换的模拟电压的输入通道。

可由 1 位或 8 位存储器操作指令设置 ADS。

复位信号产生将该寄存器清零(00H)。

图 **12-8.** 模拟输入通道选择寄存器 **(ADS)**的格式

地址: FF29H　　复位后: 00H　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADS | 0 | 0 | 0 | 0 | 0 | ADS2 | ADS1 | ADS0 |

| ADS2 | ADS1 | ADS0 | 模拟输入通道的选择 |
|---|---|---|---|
| 0 | 0 | 0 | ANI0 |
| 0 | 0 | 1 | ANI1 |
| 0 | 1 | 0 | ANI2 |
| 0 | 1 | 1 | ANI3 |
| 1 | 0 | 0 | ANI4 |
| 1 | 0 | 1 | ANI5 |
| 1 | 1 | 0 | ANI6 |
| 1 | 1 | 1 | ANI7 |

注意事项

**1.** 第 **3**～**7** 位必须清零。

**2.** 使用端口模式寄存器 **2(PM2)**设置一个用于 **A/D** 转换通道的端口为输入模式。

**3.** 如果对 **ADS** 赋值，会产生一个等待周期。当 **CPU** 工作在副系统时钟，且外围硬件时钟处于停止状态时，不要对 **ADS** 赋值。需要了解详细信息，可参阅 第三十二章 等待注意事项。

**(5) A/D 端口配置寄存器 (ADPC)**

这个寄存器用于将 ANI0/P20 ～ ANI7/P27 引脚切换为 A/D 转换器的模拟输入或者数字 I/O 端口。

可由 1 位或 8 位存储器操作指令设置 ADPC 。

复位信号产生将该寄存器清零(00H)。

**图 12-9. A/D 端口配置寄存器(ADPC)的格式**

地址: FF2FH　　复位后: 00H　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADPC | 0 | 0 | 0 | 0 | ADPC3 | ADPC2 | ADPC1 | ADPC0 |

| ADPC3 | ADPC2 | ADPC1 | ADPC0 | 模拟输入(A)/数字 I/O(D)的切换 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | ANI7/ P27 | ANI6/ P26 | ANI5/ P25 | ANI4/ P24 | ANI3/ P23 | ANI2/ P22 | ANI1/ P21 | ANI0/ P20 |
| 0 | 0 | 0 | 0 | A | A | A | A | A | A | A | A |
| 0 | 0 | 0 | 1 | A | A | A | A | A | A | A | D |
| 0 | 0 | 1 | 0 | A | A | A | A | A | A | D | D |
| 0 | 0 | 1 | 1 | A | A | A | A | A | D | D | D |
| 0 | 1 | 0 | 0 | A | A | A | A | D | D | D | D |
| 0 | 1 | 0 | 1 | A | A | A | D | D | D | D | D |
| 0 | 1 | 1 | 0 | A | A | D | D | D | D | D | D |
| 0 | 1 | 1 | 1 | A | D | D | D | D | D | D | D |
| 1 | 0 | 0 | 0 | D | D | D | D | D | D | D | D |
| 其它情况 | | | | 禁止设置 | | | | | | | |

注意事项 **1.** 用端口模式寄存器 **2(PM2)**设置一个用于 **A/D** 转换通道的端口为输入模式。

**2.** 如果将数据写入 **ADPC**，会产生一个等待周期。当 **CPU** 工作在副系统时钟，且外围硬件时钟处于停止状态时，不要将数据写入 **ADPC**。需要了解详细信息，可参阅 第三十二章 等待注意事项。

**(6) 端口模式寄存器 2 (PM2)**

在使用 ANI0/P20 ～ ANI7/P27 引脚作为模拟输入端口时，将 PM20 ～PM27 设为 1。此时 P20 ～P27 的输出锁存器的内容为 0 或 1。

如果将 PM20 ～ PM27 设为 0，则它们不能用作模拟输入端口引脚。

可由 1 位或 8 位的存储器操作指令来设置 PM2。

复位信号产生将该寄存器设置为 FFH。

**图 12-10. 端口模式寄存器 2 (PM2)的格式**

地址: FF22H　　复位后: FFH　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM2 | PM27 | PM26 | PM25 | PM24 | PM23 | PM22 | PM21 | PM20 |

| PM2n | P2n 引脚 I/O 模式的选择(n = 0～7) |
|---|---|
| 0 | 输出模式(输出缓冲器打开) |
| 1 | 输入模式(输出缓冲器关闭) |

根据 ADPC、ADS 和 PM2 的设置，ANI0/P20 ～ANI7/P27 引脚功能如下所示。

**图 12-3. ANI0/P20 ～ ANI7/P27 引脚功能的设置**

| ADPC | PM2 | ADS | ANI0/P20 ～ ANI7/P27 引脚 |
|---|---|---|---|
| 模拟输入的选择 | 输入模式 | 选择 ANI | 模拟输入 (被转换) |
| | | 不选择 ANI | 模拟输入 (不被转换) |
| | 输出模式 | 选择 ANI | 禁止设置 |
| | | 不选择 ANI | |
| 数字 I/O 的选择 | 输入模式 | — | 数字输入 |
| | 输出模式 | — | 数字输出 |

## 12.4 A/D 转换器的操作

### 12.4.1 A/D 转换器的基本操作

<1> 把 A/D 转换器模式寄存器的第 0 位(ADCE)置 1 启动比较器的操作。

<2> 通过 A/D 端口配置寄存器设置通道，用于 A/D 转换的模拟输入，并且通过端口模式寄存器 2(PM2)设置为输入模式。

<3> 根据 ADM 的第 5～1 位(FR2 ～ FR0、LV1 和 LV0)设置 A/D 转换时间。

<4> 通过模拟输入通道选择寄存器(ADS)选择一个通道用于 A/D 转换

<5> 设置 ADM 的第 7 位(ADCS)为 1，启动转换操作。
(<6> ～<12> 的操作由硬件完成)

<6> 由采样&保持电路对输入到已选中的模拟输入通道的电压进行采样。

<7> 在经过一定时间的采样后，采样&保持电路处于保持状态，且在 A/D 转换操作结束前一直保持采样电压。

<8> 设置逐次逼近寄存器(SAR)的第 9 位。通过分接选择器将串联电阻串的分接电压置为(1/2) $AV_{REF}$。

<9> 由电压比较器比较串联电阻串的分接电压与采样电压。如果模拟输入电压高于(1/2) $AV_{REF}$，则 SAR 的 MSB =1；如果模拟输入电压低于(1/2) $AV_{REF}$，则 SAR 的 MSB = 0。

<10> 接下来，SAR 的第 8 位自动置 1，并进入下一个比较过程。根据第 9 位的预置值选择串联电阻串的分接电压，具体描述如下。
- 第 9 位 = 1： (3/4) $AV_{REF}$
- 第 9 位 = 0： (1/4) $AV_{REF}$

比较分接电压与采样电压，并设置 SAR 的第 8 位，如下所示。
- 模拟输入电压 ≥ 分接电压：第 8 位 = 1
- 模拟输入电压 < 分接电压： 第 8 位 = 0

<11> 按此方式继续进行比较，直至 SAR 的第 0 位。

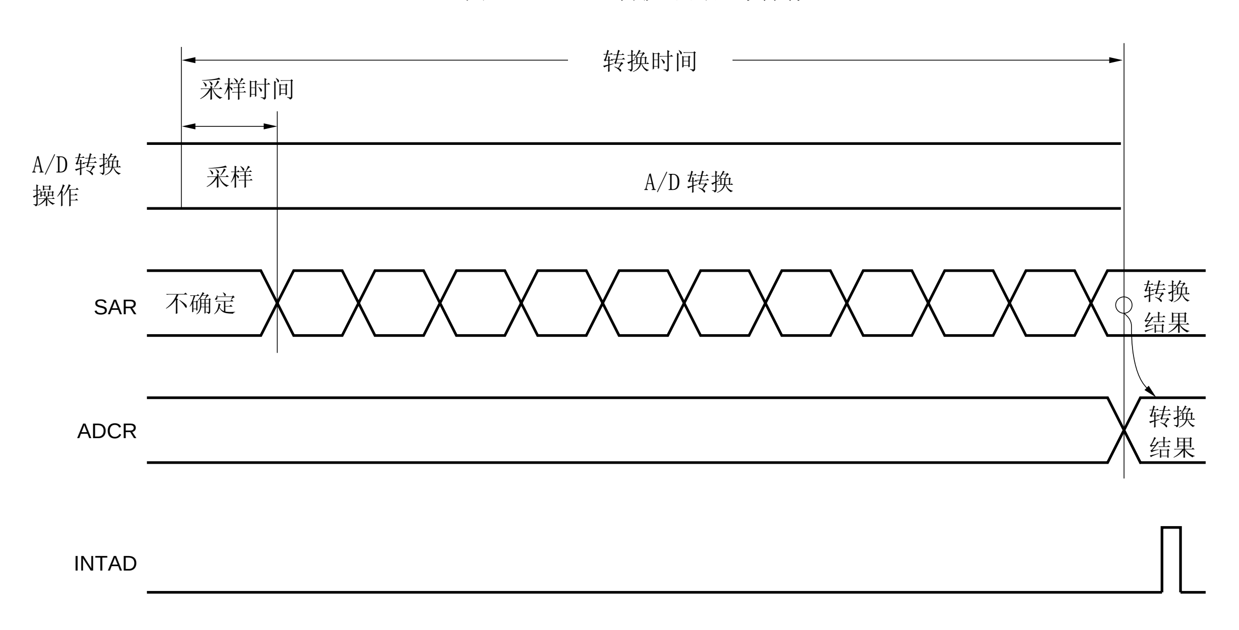
<12> 全部 10 位比较完成后，在 SAR 中保留一个有效的数值结果，然后将结果传送至 A/D 转换结果寄存器(ADCR，ADCRH)中，并锁存。
同时也会产生 A/D 转换结束中断请求 (INTAD)。

<13> 反复执行步骤 <6> ～ <12>，直至 ADCS 被清零(0)。
将 ADCS 清零，以停止 A/D 转换器操作。
当 ADCE = 1 时，若要重新启动 A/D 转换操作，应从步骤<5>开始。当 ADCE = 0 时，若要再次启动 A/D 转换操作，设置 ADCE=1，等待至少 1 μs，然后从步骤<5>开始操作。如要改变 A/D 转换的通道，则从步骤<4>开始。

**注意事项 必须确保 <1> ～ <5> 的操作时间至少为 1 μs 。**

**备注** 有两种类型的 A/D 转换结果寄存器可以使用。
- ADCR (16 位): 存储 10 位 A/D 转换值
- ADCRH (8 位): 存储 8 位 A/D 转换值

**图 12-11. A/D 转换器的基本操作**




直到用软件将 A/D 转换器模式寄存器(ADM)的第 7 位(ADCS)复位(0)，连续执行 A/D 转换操作。

在 A/D 转换期间，如果对模拟输入通道选择寄存器(ADS)进行写操作，则转换操作被初始化，并且若 ADCS 被设置为 1，则转换操作重新开始。

复位信号产生将 A/D 转换结果寄存器(ADCR，ADCRH)的内容设置为 0000H 或 00H。

### 12.4.2 输入电压和转换结果

输入到模拟输入引脚(ANI0 ～ ANI7)的模拟输入电压与理论上的 A/D 转换结果(存储在 10 位 A/D 转换结果寄存器(ADCR)中)之间的关系表示如下。

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$ADCR = SAR \times 64$$

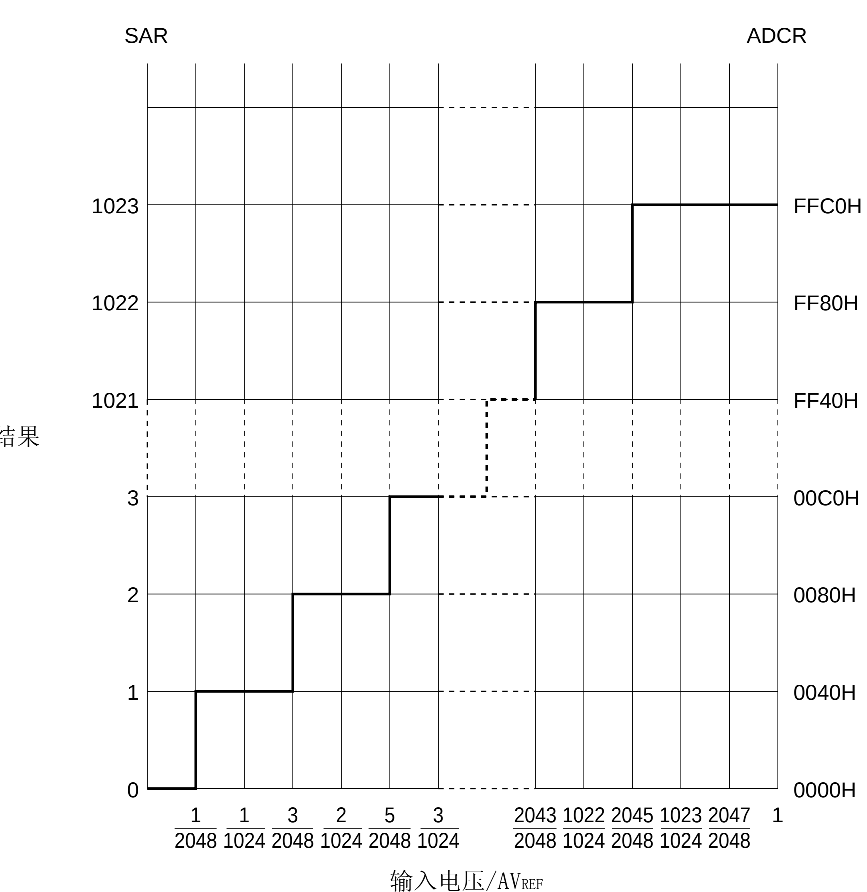
或

$$\left(\frac{ADCR}{64} - 0.5\right) \times \frac{AV_{REF}}{1024} \le V_{AIN} < \left(\frac{ADCR}{64} + 0.5\right) \times \frac{AV_{REF}}{1024}$$

其中， INT()： 该函数返回括号中值的整数部分。
$V_{AIN}$： 模拟输入电压
$AV_{REF}$： $AV_{REF}$ 引脚电压
ADCR： A/D 转换结果寄存器(ADCR)的值
SAR： 逐次逼近寄存器

图 12-12 显示模拟输入电压与 A/D 转换结果之间的关系。

**图 12-12. 模拟输入电压和 A/D 转换结果之间的关系**



### 12.4.3 A/D 转换器操作模式

A/D 转换器的操作模式为选择模式。由模拟输入通道选择寄存器(ADS)从 ANI0 ～ ANI7 中选择一个模拟输入通道，并执行 A/D 转换。

**(1) A/D 转换操作**

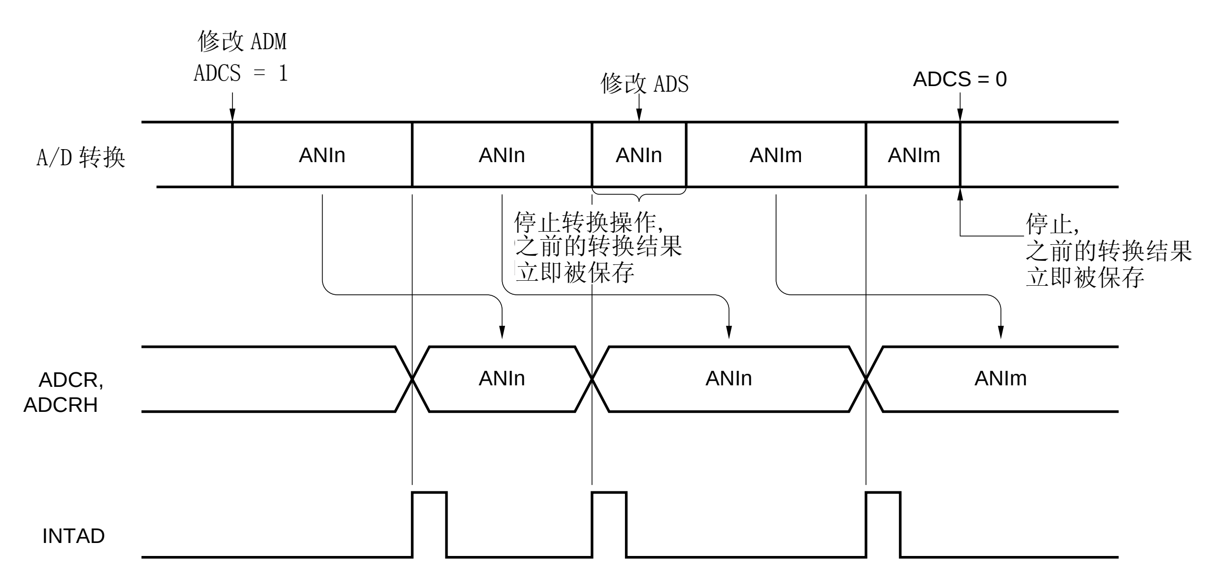
通过将 A/D 转换器模式寄存器(ADM)的第 7 位(ADCS)置 1，可启动电压的 A/D 转换操作，该电压由模拟输入通道选择寄存器(ADS)选择的模拟输入引脚输入。

当 A/D 转换完成时，将 A/D 转换结果存于 A/D 转换结果寄存器(ADCR)中，并产生一个中断请求信号(INTAD)。当一个 A/D 转换结束时，下一个 A/D 转换操作将立即开始。

如果在 A/D 转换期间 ADS 的值被重写，则停止当前正在执行的 A/D 转换操作，重新开始。

如果在 A/D 转换期间 ADCS 被置 0，则 A/D 转换立即停止。此时，之前的转换结果立即被保存。

**图 12-13. A/D 转换操作**




备注 **1.** n = 0 ～ 7
**2.** m = 0 ～ 7

设置方法描述如下。

<1> 设置 A/D 转换器模式寄存器(ADM)的第 0 位(ADCE)为 1。

<2> 根据 A/D 端口配置寄存器的第 3～0 位(ADPC3 ～ ADPC0)和端口模式寄存器 2(PM2)的第 7～0 位(PM27 ～ PM20)，设置在模拟输入模式下使用的通道。

<3> 根据 ADM 的第 5～1 位(FR2 ～ FR0、LV1 和 LV0)选择转换时间。

<4> 使用模拟输入通道选择寄存器(ADS)的第 2～0 位(ADS2 ～ ADS0)，选择通道。

<5> 将 ADM 的第 7 位 (ADCS)置 1，启动 A/D 转换操作。

<6> 当 A/D 转换结束，产生一个中断请求信号(INTAD)。

<7> 将 A/D 转换数据传送至 A/D 转换结果寄存器(ADCR，ADCRH)中。

<改变通道>

<8> 根据 ADS 的第 2～0 位(ADS2 ～ ADS0)改变通道，启动 A/D 转换操作。

<9> 当 A/D 转换结束，产生一个中断请求信号(INTAD)。

<10> 将 A/D 转换数据传送至 A/D 转换结果寄存器(ADCR，ADCRH)中。

<完成 A/D 转换>

<11> 将 ADCS 清零。

<12> 将 ADCE 清零。

**注意事项 1. 必须确保步骤 <1> ～ <5>的操作时间至少为 1 $\mu$s。**

**2. <1> 可以在<2> 和 <4>之间进行.**

**3. <1>可以被省略。但在这种情况中（<5>之后）忽略第一个转换结果。**

**4. 步骤<6> ～ <9>所经历的时间与使用 ADM 的第 5～1 位(FR2 ～ FR0，LV1，LV0)设置的转换时间不同。步骤<8> ～ <9>所经历的时间为 FR2 ～ FR0，LV1，LV0 设置的转换时间。**

## 12.5 A/D 转换器特征表的阅读方法

以下介绍 A/D 转换器中的专用术语。

**(1)** 分辨率

这是可识别的最小的模拟输入电压，即每位数字输出的模拟输入电压的百分比，称为 1LSB(最低有效位)。对于满度的 1LSB 的百分比用%FSR(满度范围)表示。

当分辨率为 10 位时 1LSB 表示如下。

$$1LSB = 1/2^{10} = 1/1024$$
$$= 0.098\%FSR$$

精确度与分辨率无关，而由总误差决定。

**(2)** 总误差

总误差是指实际测量值与理论值之间的最大误差。
零度误差、满度误差、积分线性误差和微分线性误差等组合起来表示总误差。
注意量化误差不属于特征表中总误差的范围。

**(3)** 量化误差

当模拟值转换成数字值时，通常会产生±1/2LSB 的误差。在一个 A/D 转换器中，相差±1/2LSB 的模拟输入电压被转换成相同的数字代码，因此量化误差不可避免。
注意量化误差不属于特征表中总误差、零度误差、满度误差、积分线性误差和微分线性误差的范围。

**图 12-14. 总误差**

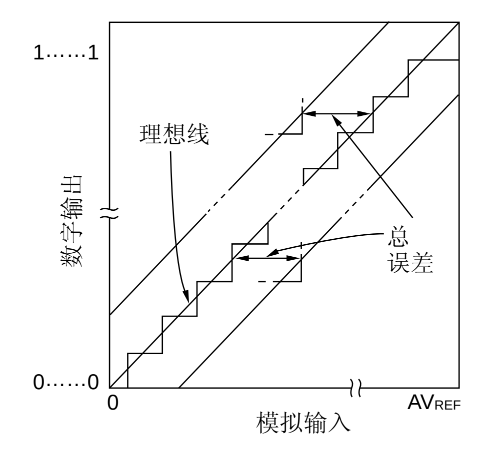
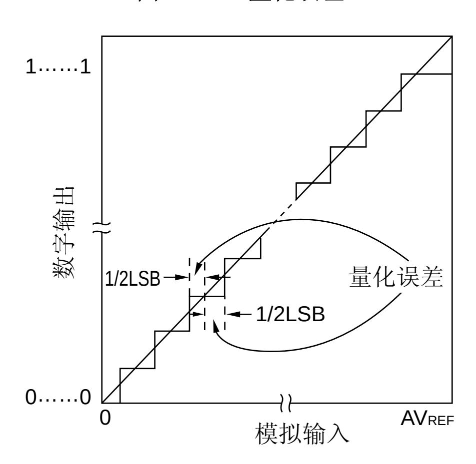



**图 12-15. 量化误差**




**(4)** 零度误差

零度误差表示当数字输出范围在 0......000 ～ 0......001 之间时模拟输入电压的实际测量值与理论值(1/2LSB)之间的误差。
如果实际测量值大于理论值，零度误差表示当数字输出范围在 0……001 ～ 0……010 之间时模拟输入电压的实际测量值与理论值(3/2LSB)之间的误差。

**(5) 满度误差**

满度误差表示当数字输出范围在 1......110 ～ 1......111 之间时模拟输入电压的实际测量值与理论值(满度 – 3/2LSB)之间的误差。

**(6) 积分线性误差**

积分线性误差说明了转换特征偏离理想线性关系的程度。它表示当零度误差和满度误差均为 0 时实际测量值与理想直线之间误差的最大值。

**(7) 微分线性误差**

当代码输出的理想宽度为 1LSB 时，微分线性误差表示实际测量值与理想值之间的差距。

**图 12-16. 零度误差**

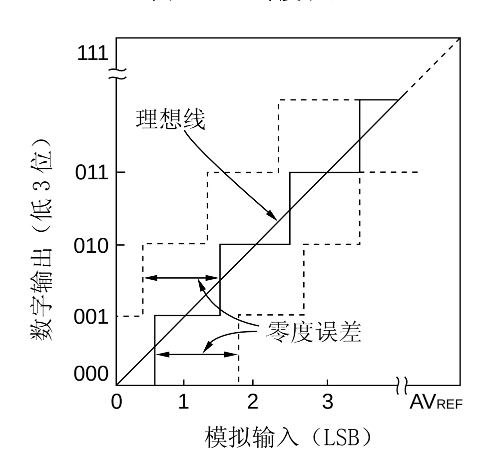



**图 12-17. 满度误差**

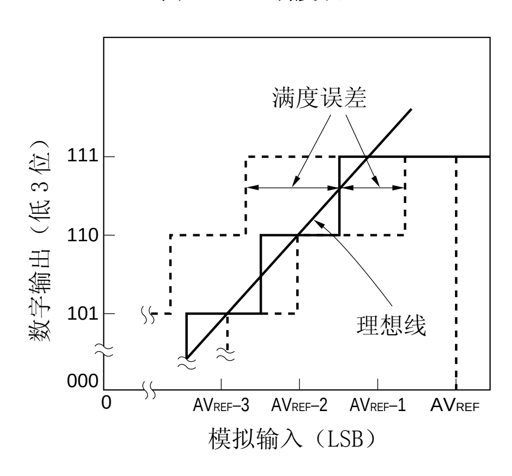



**图 12-18. 积分线性误差**

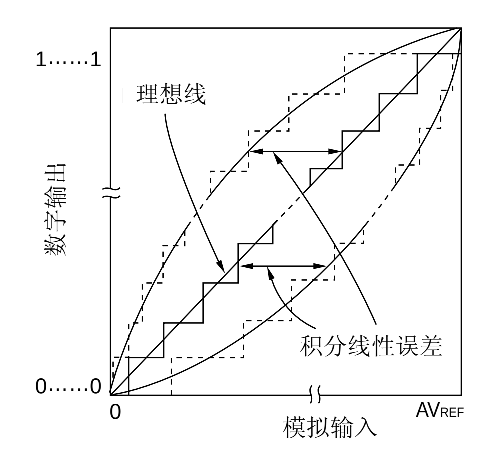
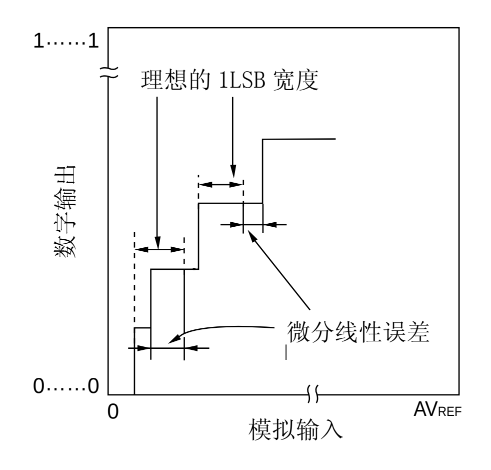



**图 12-19. 微分线性误差**



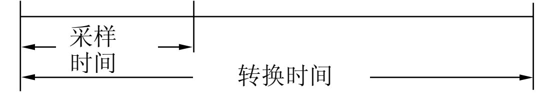

**(8) 转换时间**

转换时间表示从开始采样到获取数字输出所经历的时间。

采样时间包含在特征表中的转换时间中。

**(9) 采样时间**

采样时间表示模拟电压的模拟开关被打开到模拟电压被采样&保持电路采样所需的时间。



## 12.6 A/D 转换器使用注意事项

**(1) STOP 模式中的操作电流**

在 STOP 模式中 A/D 转换器停止操作。此时，将 A/D 转换器模式寄存器(ADM)的第 7 位(ADCS)和第 0 位(ADCE)清零，可以降低操作电流。

若要从待机状态重新启动，将中断请求标志寄存器 1L (IF1L)的第 0 位(ADIF)清零(0)，然后开始操作。

**(2) ANI0 ～ ANI7 的输入范围**

观察 ANI0 ～ ANI7 输入电压的额定范围。如果输入到模拟输入通道的电压大于等于 $AV_{REF}$，或者小于等于 $AV_{SS}$(即使在绝对最大额定范围之内)，则该通道的转换值不确定。此外，其它通道的转换值也可能会受影响。

**(3) 冲突操作**

<1> 转换结束后，通过指令对 A/D 转换结果寄存器(ADCR，ADCRH)的写操作和 ADCR 或 ADCRH 的写操作之间的冲突。

ADCR 或 ADCRH 读操作的优先级高。在执行读操作后，才将新的转换结果写入 ADCR 或 ADCRH。

<2> 转换结束后，ADCR 或 ADCRH 的写操作、A/D 转换器模式寄存器(ADM)的写操作以及模拟输入通道选择寄存器(ADS)或 A/D 端口配置寄存器(ADPC)的写操作之间的冲突。

ADM、ADS 或 ADPC 的写操作的优先级高。不执行 ADCR 或 ADCRH 的写操作，也不会产生转换结束中断信号(INTAD)。

**(4) 解决噪音问题的方法**

为了保持 10 位分辨率，必须注意输入到 $AV_{REF}$ 引脚和 ANI0～ANI7 引脚的噪音。

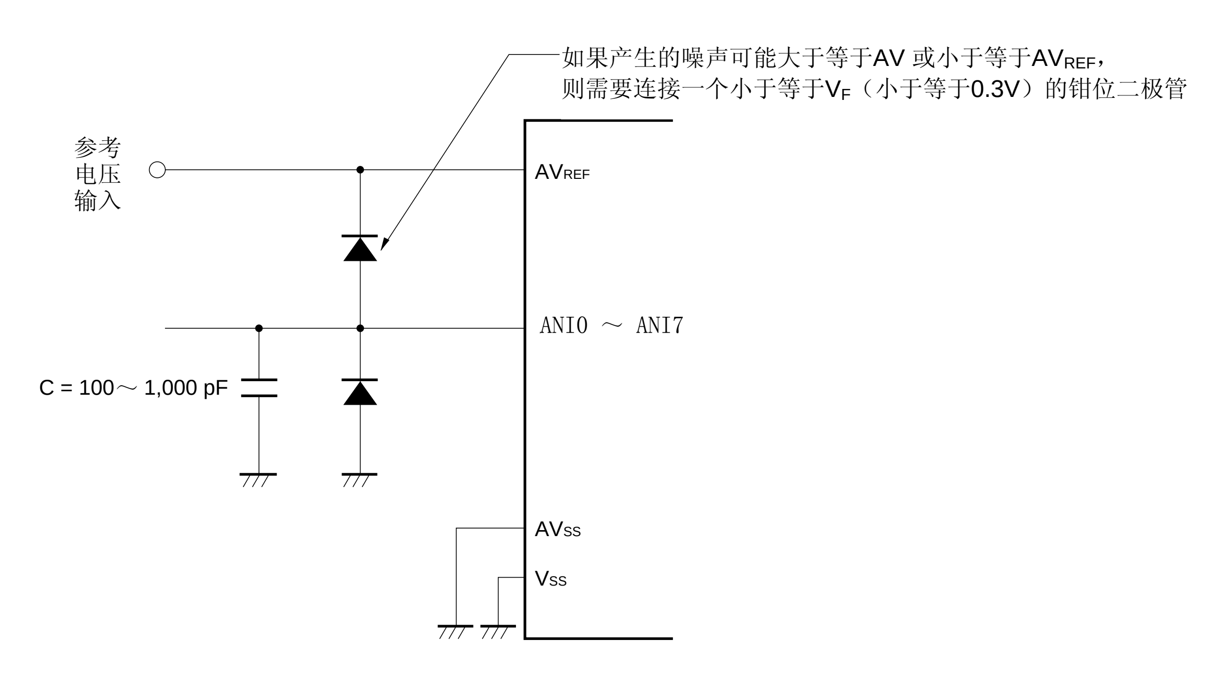
<1> 连接一个低等效电阻和优质频率响应的电容到电源上

<2> 模拟输入源的输出阻抗越大，干扰就越大。为了降低噪音，建议按图 13-20 所示连接外部 C。

<3> 在转换过程中不要切换引脚。

<4> 如果在转换开始后立即设置 HALT 模式，则可以改善精度。

**图 12-20. 模拟输入引脚连接**




**(5) ANI0/P20 ～ ANI7/P27**

<1> 模拟输入引脚(ANI0 ～ ANI7)也可用作输入端口引脚(P20 ～ P27)。
当选择 ANI0 ～ ANI7 中的任意一个通道执行 A/D 转换时，转换过程中不要访问 P20 ～ P27；否则转换分辨率可能会降低。建议从 ANI0/P20(离 $AV_{REF}$ 最远)开始，选择引脚用作 P20 ～ P27。

<2> 如果正在进行 A/D 转换的引脚的相邻引脚有数字脉冲，则由于噪音耦合，有可能得不到预期的 A/D 转换值。因此在进行 A/D 转换时不要在相邻引脚引用脉冲。

**(6) ANI0 ～ ANI7 引脚的输入阻抗**

采样期间 A/D 转换器对采样电容充电，以便进行采样。

因此当不进行采样时仅有漏电流经过，而在采样期间则有为电容充电的电流，因此根据是否进行采样，输入阻抗会波动，且无法解决。

为了使采样有效，建议模拟输入源的输出阻抗小于等于 10 kΩ，并将一个 100 pF 左右 的电容连接到 ANI0 ～ ANI7 引脚上(参见 **图 12-20**)。

**(7) $AV_{REF}$ 引脚输入阻抗**

在 $AV_{REF}$ 与 $AV_{SS}$ 引脚之间连接几十千欧的串联电阻串。

因此，如果参考电压源的输出阻抗很高，当它串联连接到 $AV_{REF}$ 与 $AV_{SS}$ 引脚之间的串联电阻串时，会导致较大的参考电压误差。

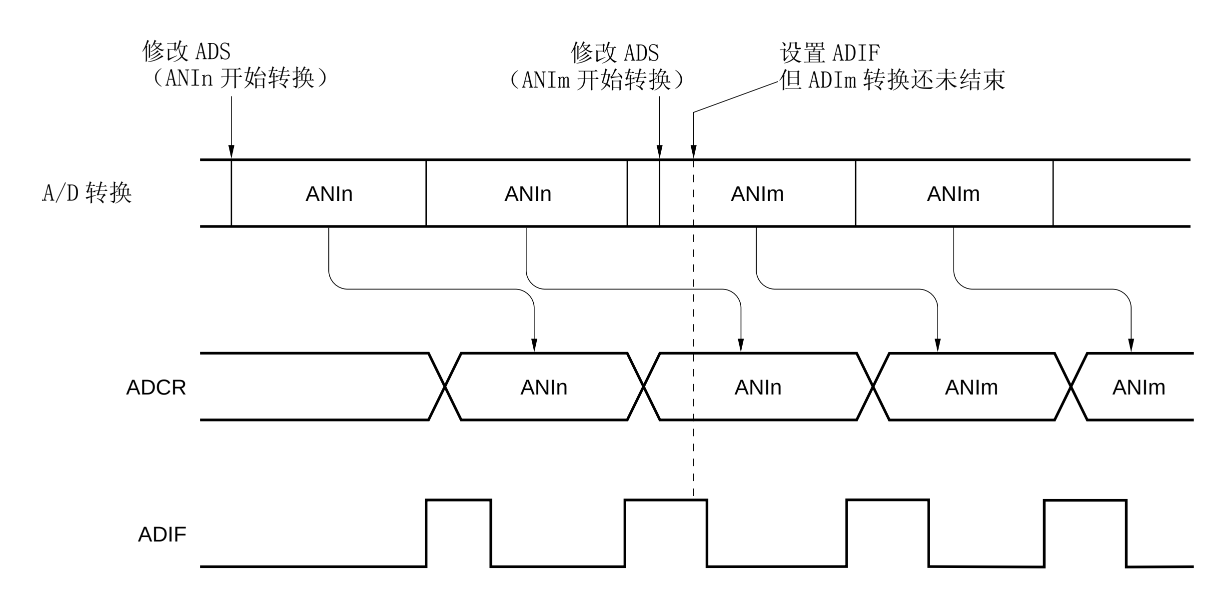
**(8) 中断请求标志(ADIF)**

即使模拟输入通道选择寄存器(ADS)的值被修改，中断请求标志(ADIF)也不会被清零。

因此，如果在 A/D 转换期间有一个模拟输入引脚发生变化，则在 ADS 被修改之前，模拟输入通道的 A/D 转换结果和改变前的模拟通道的 ADIF 可能被设置。此时需要注意是，当 ADS 修改后立即读取 ADIF 时，即使修改后的模拟输入的 A/D 转换尚未结束，也会设置 ADIF。

当 A/D 转换停止后又重新开始时，在启动前先对 ADIF 清零。

**图 12-21. A/D 转换结束中断请求产生的时序**




备注 **1.** n = 0 ～ 7
**2.** m = 0 ～ 7

**(9) A/D 转换刚开始时的转换结果**

在 A/D 转换开始后，若在 ADCE 置 1 后的 1 $\mu$s 内对 ADCS 置 1，或者 ADCE=0 时 ADCS 置 1，那么第一次的 A/D 转换值可能不在额定范围内。可采取措施，如悬挂 A/D 转换结束中断请求(INTAD)，并删除第 1 次转换结果。

**(10)A/D 转换结果寄存器(ADCR, ADCRH)的读取操作**

当对 A/D 转换器模式寄存器(ADM)、模拟输入通道选择寄存器(ADS)和 A/D 端口配置寄存器(ADPC)进行写操作时，ADCR 和 ADCRH 的内容可能不确定。在对 ADM、ADS 和 ADPC 进行写操作前且转换结束后读取转换结果。如果采用与上述不同的时序操作可能会读取到不正确的转换结果。

**(11)**内部等效电路

模拟输入模块的等效电路如下所示。

**图 12-22. ANIn** 引脚的内部等效电路

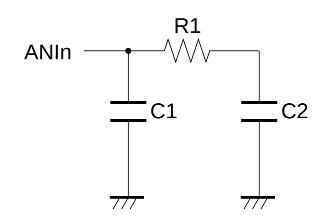



**表 12-4.** 等效电路的电阻和电容 **(**参考值**)**

| $AV_{REF}$ | R1 | C1 | C2 |
|---|---|---|---|
| 4.0 V ≤ $AV_{REF}$ ≤ 5.5 V | 8.1 kΩ | 8 pF | 5 pF |
| 2.7 V ≤ $AV_{REF}$ < 4.0 V | 31 kΩ | 8 pF | 5 pF |
| 2.3 V ≤ $AV_{REF}$ < 2.7 V | 381 kΩ | 8 pF | 5 pF |

备注　1. 表 12-4 所显示的电阻和电容值仅为参考值。

2. n = 0 ～ 7

## 13.1 串行接口 UART0 的功能

串行接口 UART0 有以下两种模式。

**(1)** 操作停止模式
这个模式在不进行串口通信时使用，以降低功耗。
需要了解详细信息，请参阅 **13.4.1 操作停止模式。**

**(2)** 异步串行接口(UART)模式
该模式的功能概括如下。
需要了解详细信息，请参阅 **13.4.2 异步串行接口(UART)模式** 和 **13.4.3** 专用波特率发生器。

<R>
- 最大传输率： 625 kbps
- 双引脚配置　　TXD0: 发送数据输出引脚
  　　　　　　　RxD0: 接收数据输入引脚
- 可选择通信数据的宽度为 7 或 8 位。
- 专用内置 5 位波特率发生器，允许设置任意大小的波特率。
- 可分别执行发送和接收操作(全双工操作)。
- 恒为 LSB-first 通信

注意事项 **1.** 如果串行接口 **UART0** 使用的时钟未停止**(**例如处于 **HALT** 模式下**)**，则可继续正常操作。如果时钟已停止**(**例如处于 **STOP** 模式下**)**，则每个寄存器都停止操作，并且在时钟停止之前及时保存数据。**TxD0** 引脚在时钟停止之前立即保存数据并输出。但在时钟恢复使用后的操作不能得到保证。因此必须复位电路，使 **POWER0 = 0**、**RXE0 = 0** 以及 **TXE0 = 0**。

**2.** 先设置 **POWER0 = 1**，然后设置 **TXE0 = 1(**发送**)**或 **RXE0 = 1(**接收**)**开始通信。

**3.** 由 **BRGC0** 设置基本时钟**($f_{XCLK0}$)**，可使 **TXE0** 与 **RXE0** 同步。为了能够再次发送或接收，在 **TXE0** 或 **RXE0** 被清零后，至少要经历两个基本时钟再将 **TXE0** 或 **RXE0** 置 **1**。如果在不到两个基本时钟时设置 **TXE0** 或 **RXE0**，则发送或接收电路可能不被初始化。

**4.** 设置 **TXE0=1** 之后经过至少一个基本时钟**($f_{XCLK0}$)**才能发送数据到 **TXS0**。

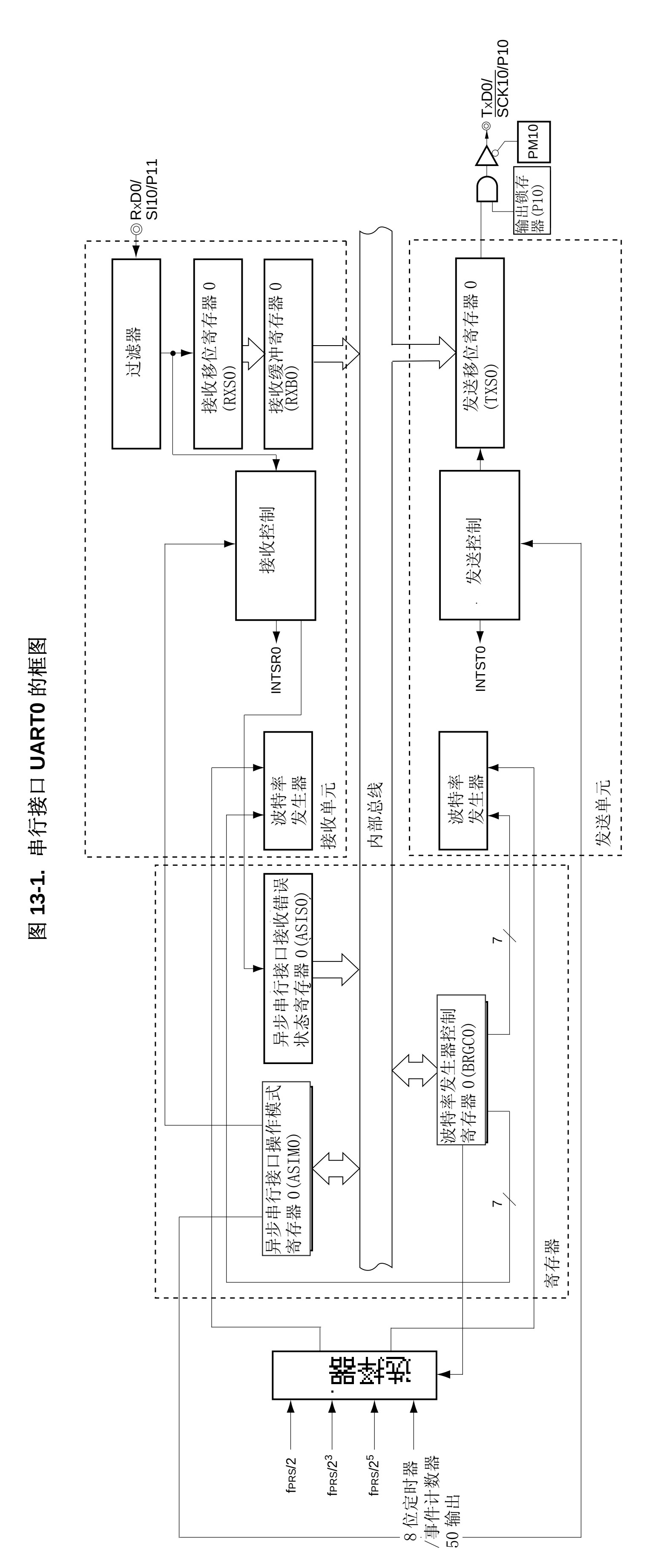
## 13.2 串行接口 UART0 的配置

串行接口 UART0 包括以下硬件。

**表 13-1. 串行接口 UART0 的配置**

| 项目 | 配置 |
|---|---|
| 寄存器 | 接收缓冲寄存器 0 (RXB0)<br>接收移位寄存器 0 (RXS0)<br>发送移位寄存器 0 (TXS0) |
| 控制寄存器 | 异步串行接口操作模式寄存器 0 (ASIM0)<br>异步串行接口接收错误状态寄存器 0 (ASIS0)<br>波特率发生器控制寄存器 0 (BRGC0)<br>端口模式寄存器 1 (PM1)<br>端口寄存器 1 (P1) |

**图 13-1. 串行接口 UART0 的框图**



**(1) 接收缓冲寄存器 0 (RXB0)**

该 8 位寄存器用于存储由接收移位寄存器 0(RXS0)转换的并行数据。

每接收到 1 字节的数据，新的接收数据就从接收移位寄存器 0(RXS0)被传送到该寄存器中。

如果数据宽度被设定为 7 位，则将接收到的数据传送到 RXB0 的 0 ～ 6 位，而 MSB 位恒为 0。

如果出现溢出错误(OVE0)，则不把接收数据传送到 RXB0 中。

可由 8 位存储器操作指令读取 RXB0 的内容。不能将数据写入该寄存器中。

复位信号的产生和 POWER0 = 0 将该寄存器设置为 FFH。

**(2) 接收移位寄存器 0 (RXS0)**

该寄存器将输入到 RxD0 引脚的串行数据转换成并行数据。

程序不能直接使用 RXS0。

**(3) 发送移位寄存器 0 (TXS0)**

该寄存器用于设置发送数据。当数据写入 TXS0 时启动发送操作，串行数据从 TxD0 引脚输出。

可由 8 位存储器操作指令对 TXS0 进行写操作。该寄存器的内容不能被读取。

复位信号的产生、POWER0 = 0 和 TXE0 = 0 将该寄存器设置为 FFH。

注意事项　**1. 设置 TXE0=1 之后经过至少一个基本时钟($f_{XCLK0}$)才能发送数据到 TXS0。**

**2. 在发送完成中断信号(INTST0)产生之前，不要把下一个发送数据写入 TXS0。**

## 13.3 控制串行接口 UART0 的寄存器

串行接口 UART0 由以下 5 个寄存器控制。

- 异步串行接口操作模式寄存器 0 (ASIM0)
- 异步串行接口接收错误状态寄存器 0 (ASIS0)
- 波特率发生器控制寄存器 0 (BRGC0)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

**(1) 异步串行接口操作模式寄存器 0 (ASIM0)**

该 8 位寄存器用于控制串行接口 UART0 的串行通信操作。

该寄存器可由 1 位或 8 位存储器操作指令设置。

复位信号的产生将该寄存器设置为 01H。

**图 13-2. 异步串行接口操作模式寄存器 0 (ASIM0)的格式 (1/2)**

地址: FF70H 复位后: 01H R/W

| 符号 | <7> | <6> | <5> | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ASIM0 | POWER0 | TXE0 | RXE0 | PS01 | PS00 | CL0 | SL0 | 1 |

| POWER0 | 内部操作时钟的允许/禁止操作 |
|---|---|
| 0[注1] | 禁止内部操作时钟的操作 (时钟恒为低电平) 并异步复位内部电路[注2] |
| 1 | 允许内部操作时钟的操作 |

| TXE0 | 允许/禁止发送 |
|---|---|
| 0 | 禁止发送 (同步复位发送电路) |
| 1 | 允许发送 |

| RXE0 | 允许/禁止接收 |
|---|---|
| 0 | 禁止接收 (同步复位接收电路) |
| 1 | 允许接收 |

注　**1.** 当 POWER0 = 0 时，从 RxD0 引脚输入的信号恒为高电平。

　　**2.** 对异步串行接口接收错误状态寄存器 0(ASIS0)、发送移位寄存器 0(TXS0)和接收缓冲寄存器 0(RXB0)进行复位。

**图 13-2. 异步串行接口操作模式寄存器 0 (ASIM0)的格式 (2/2)**

| PS01 | PS00 | 发送操作 | 接收操作 |
| --- | --- | --- | --- |
| 0 | 0 | 不输出校验位 | 无校验接收 |
| 0 | 1 | 输出零校验 | 零校验接收[注] |
| 1 | 0 | 输出奇校验 | 按奇校验判断 |
| 1 | 1 | 输出偶校验 | 按偶校验判断 |

| CL0 | 指定发送/接收数据的字符宽度 |
| --- | --- |
| 0 | 数据的字符宽度 = 7 位 |
| 1 | 数据的字符宽度 = 8 位 |

| SL0 | 指定发送数据停止位的个数 |
| --- | --- |
| 0 | 停止位个数 = 1 |
| 1 | 停止位个数 = 2 |

注　如果选择“按零校验接收”，则无法判断校验。因此不要设置异步串行接口接收错误状态寄存器 0(ASIS0)的第 2 位(PE0)，这样就不会产生错误中断。

注意事项
1. 启动发送时，对 **POWER0** 置 **1**，然后对 **TXE0** 置 **1**。停止发送时，先将 **TXE0** 清零，然后将 **POWER0** 清零。
2. 启动接收时，对 **POWER0** 置 **1**，然后对 **RXE0** 置 **1**。停止接收时，先将 **RXE0** 清零，然后将 **POWER0** 清零。
3. 当 **RxD0** 引脚输入为高电平时，先对 **POWER0** 置 **1**，再对 **RXE0** 置 **1**。当输入为低电平，**POWER0** 置 **1**，**RXE0** 置 **1** 时，启动接收操作。
4. 通过 **BRGC0** 设置的基本时钟(**$f_{XCLK0}$**)同步 **TXE0** 与 **RXE0**。为了能够再次发送或接收，在 **TXE0** 或 **RXE0** 被清零后，至少要经历两个基本时钟再将 **TXE0** 或 **RXE0** 置 **1**。如果在不到两个基本时钟的时间内设置 **TXE0** 或 **RXE0**，则发送或接收电路可能不被初始化。
5. 设置 **TXE0=1** 之后经过至少一个基本时钟(**$f_{XCLK0}$**)才能设置发送数据到 **TXS0**。
6. 在对 **PS01**、**PS00** 和 **CL0** 位重写之前，先将 **TXE0** 和 **RXE0** 位清零。
7. 在重写 **SL0** 位时，必须确保 **TXE0 = 0**。总是以“停止位的个数**=1**”的格式执行接收操作，因此不会受到 **SL0** 设置值的影响。
8. 第 **0** 位必须设置为 **1**。

**(2)　异步串行接口接收错误状态寄存器 0 (ASIS0)**

该寄存器用于指示串行接口 UART0 在接收操作完成时出现的错误状态，包括 3 个错误标志位(PE0, FE0, OVE0)。可由 8 位存储器操作指令对该寄存器进行只读操作。

复位信号或将 ASIM0 的第 5 位(RXE0)或第 7 位(POWER0)置 0，将使 ASIS0 寄存器清零为 00H。读取寄存器的值为 00H。如果产生一个接收错误,则先读取 ASIS0 再读取接收缓冲寄存器 0(RXB0)以便将出错标志位清零。

**图 13-3.　异步串行接口接收错误状态寄存器 0 (ASIS0)的格式**

地址: FF73H　复位后: 00H　R

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ASIS0 | 0 | 0 | 0 | 0 | 0 | PE0 | FE0 | OVE0 |

| PE0 | 指示校验错误的状态标志位 |
|---|---|
| 0 | 如果 POWER0 = 0 且 RXE0 = 0、或者如果读取 ASIS0 |
| 1 | 如果发送数据的校验位与接收完成时的校验位不相等 |

| FE0 | 指示帧错误的状态标志位 |
|---|---|
| 0 | 如果 POWER0 = 0 或 RXE0 = 0、或者如果读取 ASIS0 |
| 1 | 如果接收完成时未检测到停止位 |

| OVE0 | 指示溢出错误的状态标志位 |
|---|---|
| 0 | 如果 POWER0 = 0 且 RXE0 = 0、或者如果读取 ASIS0 |
| 1 | 如果接收数据被设置在 RXB0 寄存器中且在读取该数据前下一个接收操作已完成 |

注意事项

1. **根据异步串行接口操作模式寄存器 0(ASIM0)的 PS01 和 PS00 位的内容的不同，对 PE0 位的操作也有所不同。**
2. **无论停止位的个数是多少，只能将接收数据的第 1 位作为停止位进行检测。**
3. **如果出现溢出错误，则下一个接收数据不写入接收缓冲寄存器 0(RXB0)中，而是被忽略。**
4. **如果从 ASIS0 读取数据，会产生一个等待周期。当 CPU 采用副系统时钟操作而停止外围硬件时钟时，不要从 ASIS0 读取数据。需要了解详细信息，可参见 第三十二章 等待注意事项。**

**(3) 波特率发生器控制寄存器 0 (BRGC0)**

该寄存器用于选择串行接口 UART0 的基本时钟和 5 位计数器的分频值。

可由 8 位存储器操作指令设置 BRGC0。

复位信号的产生将该寄存器设置为 1FH。

**图 13-4. 波特率发生器控制寄存器 0 (BRGC0)的格式**

地址: FF71H 复位后: 1FH R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| BRGC0 | TPS01 | TPS00 | 0 | MDL04 | MDL03 | MDL02 | MDL01 | MDL00 |

| TPS01 | TPS00 | 基本时钟 ($f_{XCLK0}$) 的选择 | | | | |
|---|---|---|---|---|---|---|
| | | | $f_{PRS}$ = 2 MHz | $f_{PRS}$ = 5 MHz | $f_{PRS}$ = 10 MHz | $f_{PRS}$ = 20 MHz |
| 0 | 0 | TM50 输出注 2 | | | | |
| 0 | 1 | $f_{PRS}/2$ | 1 MHz | 2.5 MHz | 5 MHz | 10 MHz |
| 1 | 0 | $f_{PRS}/2^3$ | 250 kHz | 625 kHz | 1.25 MHz | 2.5 MHz |
| 1 | 1 | $f_{PRS}/2^5$ | 62.5 kHz | 156.25 kHz | 312.5 kHz | 625 kHz |

| MDL04 | MDL03 | MDL02 | MDL01 | MDL00 | k | 5 位计数器输出时钟的选择 |
|---|---|---|---|---|---|---|
| 0 | 0 | × | × | × | × | 禁止设置 |
| 0 | 1 | 0 | 0 | 0 | 8 | $f_{XCLK0}/8$ |
| 0 | 1 | 0 | 0 | 1 | 9 | $f_{XCLK0}/9$ |
| 0 | 1 | 0 | 1 | 0 | 10 | $f_{XCLK0}/10$ |
| • | • | • | • | • | • | • |
| 1 | 1 | 0 | 1 | 0 | 26 | $f_{XCLK0}/26$ |
| 1 | 1 | 0 | 1 | 1 | 27 | $f_{XCLK0}/27$ |
| 1 | 1 | 1 | 0 | 0 | 28 | $f_{XCLK0}/28$ |
| 1 | 1 | 1 | 0 | 1 | 29 | $f_{XCLK0}/29$ |
| 1 | 1 | 1 | 1 | 0 | 30 | $f_{XCLK0}/30$ |
| 1 | 1 | 1 | 1 | 1 | 31 | $f_{XCLK0}/31$ |

**注** 选择 TM50 输出作为基本时钟时，应注意以下几点。

- TM50 与 CR50 相等时计数时钟清零并启动模式(TMC506 = 0)

  先启动 8 位定时器/事件计数器 50，然后允许定时器 F/F 进行反转操作(TMC501 = 1)。

- PWM 模式 (TMC506 = 1)

  先启动 8 位定时器/事件计数器 50，然后设置计数时钟，使占空比 = 50%。

不论何种模式，都不必将 TO50 作为定时器输出引脚。

**注意事项　1.　在重写 MDL04 ～ MDL00 位时，必须确保 ASIM0 的第 6 位(TXE0)和第 5 位(RXE0) = 0。**

**2.　波特率为 5 位计数器输出时钟的 1/2。**

备注　1.　$f_{XCLK0}$:　根据 TPS01 和 TPS00 位选择的基本时钟的频率

2.　$f_{PRS}$:　外围硬件时钟频率

3.　k:　由 MDL04 ～ MDL00 位设置的值 (k = 8, 9, 10, ..., 31)

4.　×:　不必考虑

5.　TMC506：8 位定时器模式控制寄存器 50(TMC50)的第 6 位
　　TMC501：TMC50 的第 1 位

**(4)　端口模式寄存器 1 (PM1)**

该寄存器按位设置端口 1 的输入/输出模式。

当 P10/TxD0/$\overline{SCK10}$ 引脚用于串行接口数据输出时，将 PM10 清零且对 P10 的输出锁存器置 1。

当 P11/RxD0/SI10 引脚用于串行接口数据输入时，将 PM11 置 1。此时 P11 的输出锁存器的值可以等于 0 或 1。

可由 1 位或 8 位存储器操作指令设置 PM1。

复位信号的产生将该寄存器设置为 FFH。

**图 13-5.　端口模式寄存器 1(PM1)的格式**

地址: FF21H　复位后: FFH　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM1 | PM17 | PM16 | PM15 | PM14 | PM13 | PM12 | PM11 | PM10 |

| PM1n | P1n 引脚 I/O 模式选择 (n = 0～7) |
|---|---|
| 0 | 输出模式 (输出缓冲器打开) |
| 1 | 输入模式 (输出缓冲器关闭) |

## 13.4 串行接口 UART0 的操作

串行接口 UART0 有以下两种模式。

- 操作停止模式
- 异步串行接口(UART)模式

### 13.4.1 操作停止模式

在此模式下，不能进行串行通信，这样可降低功耗。此外，引脚可作为通用端口引脚使用。通过将 ASIM0 的第 7、6 和 5 位(POWER0、TXE0 和 RXE0)清零，可设置操作停止模式。

**(1) 使用的寄存器**

由异步串行接口操作模式寄存器 0(ASIM0)设置操作停止模式。

可由 1 位或 8 位存储器操作指令设置 ASIM0。

复位信号的产生将该寄存器设置为 01H。

地址: FF70H 复位后: 01H R/W

| 符号 | <7> | <6> | <5> | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ASIM0 | POWER0 | TXE0 | RXE0 | PS01 | PS00 | CL0 | SL0 | 1 |

| POWER0 | 内部操作时钟的允许/禁止操作 |
|---|---|
| 0[注 1] | 禁止内部操作时钟的操作(时钟恒为低电平)且对内部电路进行异步复位[注 2] |

| TXE0 | 允许/禁止发送 |
|---|---|
| 0 | 禁止发送(同步复位发送电路) |

| RXE0 | 允许/禁止接收 |
|---|---|
| 0 | 禁止接收(同步复位接收电路) |

注　**1.** 当 POWER0 = 0 时，从 RxD0 引脚输入的信号恒为高电平。

　　**2.** 对异步串行接口接收错误状态寄存器 0(ASIS0)、发送移位寄存器 0(TXS0)和接收缓冲寄存器 0(RXB0)进行复位。

**注意事项　TXE0 和 RXE0 清零后再对 POWER0 清零，可设置操作停止模式。**

**如果要启动发送或接收，设置 POWER0=1，然后设置 TXE0 或 RXE0=1。**

**备注**　如果要将 RxD0/SI10/P11 和 TxD0/$\overline{\text{SCK10}}$/P10 作为通用端口引脚使用，可参见 **第四章　端口功能**。

### 13.4.2 异步串行接口 (UART)模式

此模式下，在起始位后的 1 字节数据被发送/接收，可执行全双工操作。
可使用专用 UART 波特率发生器，这样通信选择的波特率范围较大。

**(1) 使用的寄存器**

- 异步串行接口操作模式寄存器 0 (ASIM0)
- 异步串行接口接收错误状态寄存器 0 (ASIS0)
- 波特率发生器控制寄存器 0 (BRGC0)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

在 UART 模式中设置操作的基本过程如下。

<1>　设置 BRGC0 寄存器 (参见 图 **13-4**)。
<2>　设置 ASIM0 的 1～4 位 (SL0、CL0、PS00 和 PS01)(参见 图 **13-2**)。
<3>　将 ASIM0 的第 7 位(POWER0)置 1。
<4>　将 ASIM0 的第 6 位 (TXE0)置 1。 → 允许发送。
　　　将 ASIM0 的第 5 位 (RXE0)置 1。 → 允许接收。
<5>　对 TXS0 寄存器赋值。 → 开始发送数据。

**注意事项　在设置端口模式寄存器和端口寄存器时，要考虑与通信另一方的关系。**

寄存器设置与引脚之间的关系如下所示。

**表 13-2. 寄存器设置与引脚之间的关系**

| POWER0 | TXE0 | RXE0 | PM10 | P10 | PM11 | P11 | UART0 操作 | 引脚功能 | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | TxD0/$\overline{\text{SCK10}}$/P10 | RxD0/SI10/P11 |
| 0 | 0 | 0 | ×[注] | ×[注] | ×[注] | ×[注] | 停止 | $\overline{\text{SCK10}}$/P10 | SI10/P11 |
| 1 | 0 | 1 | ×[注] | ×[注] | 1 | × | 接收 | $\overline{\text{SCK10}}$/P10 | RxD0 |
| | 1 | 0 | 0 | 1 | ×[注] | ×[注] | 发送 | TxD0 | SI10/P11 |
| | 1 | 1 | 0 | 1 | 1 | × | 发送/接收 | TxD0 | RxD0 |

**注** 能够作为端口或串行接口 CSI10 使用。

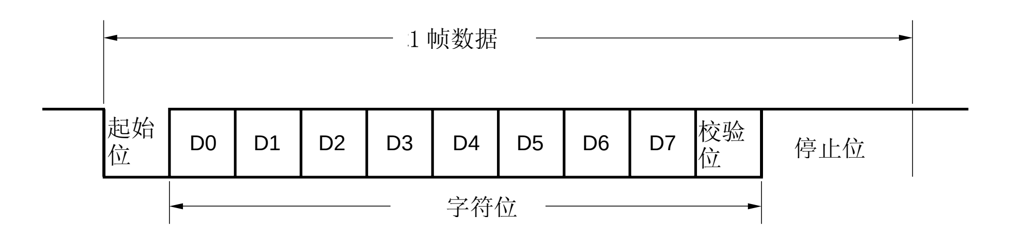
**备注**　×:　不必考虑
　POWER0: 异步串行接口操作模式寄存器 0 (ASIM0)的第 7 位
　TXE0:　ASIM0 的第 6 位
　RXE0:　ASIM0 的第 5 位
　PM1×:　端口模式寄存器
　P1×:　端口输出锁存器

**(2) 通信操作**

**(a)正常发送/接收数据的格式和波形示例**

图 13-6 和 13-7 为正常发送/接收数据的格式和波形示例。

**图 13-6. UART 正常发送/接收数据的格式**



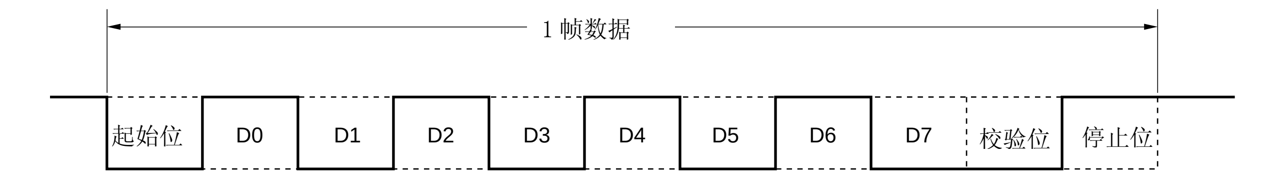

一个数据帧由以下各位组成。

- 起始位 ... 1 位
- 字符位 ... 7 或 8 位 (LSB-first)
- 校验位 ... 偶校验、奇校验、零校验或无校验
- 停止位 ... 1 或 2 位

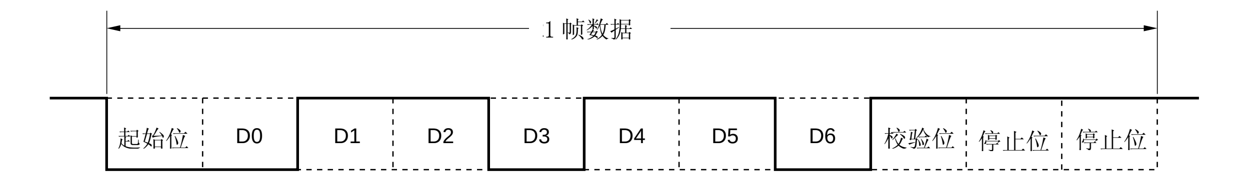
由异步串行接口操作模式寄存器 0(ASIM0)规定一个数据帧中字符位的宽度、校验方式和停止位的宽度。

**图 13-7. UART 正常发送/接收数据波形示例**

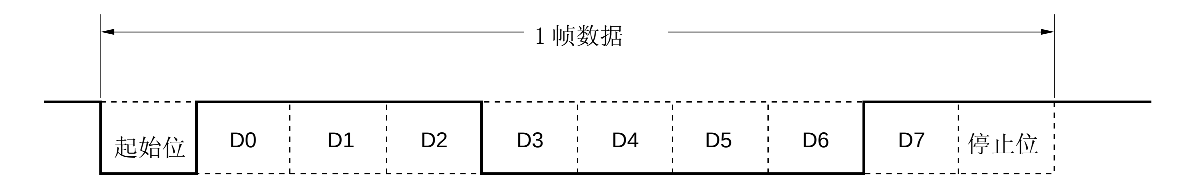
**1. 数据宽度: 8 位， 校验位: 偶校验， 停止位: 1 位， 通信数据: 55H**




**2. 数据宽度: 7 位， 校验位: 奇校验， 停止位: 2 位，通信数据: 36H**




**3. 数据宽度: 8 位， 校验位: 无， 停止位: 1 位，通信数据: 87H**



**(b) 校验方式与操作**

校验位用于检测数据通信中的位错误。通常在发送和接收两端可采用相同的校验方式。当采用偶校验和奇校验时，可检测到 1 位(奇数)错误。而采用零校验和无校验时，则不能检测到错误。

**(i) 偶校验**

- 发送

  控制发送数据，包括校验位，使得数据中“1”的个数为偶数。
  校验位的取值如下。

  如果发送数据有奇数个 “1”: 1
  如果发送数据有偶数个 “1”: 0

- 接收

  计算接收数据(包括校验位)中“1”的个数。如果结果为奇数，则产生校验错误。

**(ii) 奇校验**

- 发送

  与偶校验方式不同，奇校验用于控制发送数据，包括校验位，使得数据中“1”的个数为奇数。
  校验位取值如下。

  如果发送数据有奇数个 “1”: 0
  如果发送数据有偶数个 “1”: 1

- 接收

  计算接收数据(包括校验位)中“1”的个数。如果结果为偶数，则产生校验错误。

**(iii) 零校验**

无论发送数据为何值，发送时校验位被清零。
接收数据时不检测校验位。因此无论校验位为何值，都不会产生校验错误。

**(iv) 无校验**

发送数据中没有校验位。
进行接收操作时认为接收数据中没有校验位。因此也不会产生校验错误。

**(c)** 发送

当异步串行接口操作模式寄存器 0(ASIM0)的第 7 位(POWER0)=1 且 ASIM0 的第 6 位(TXE0)=1 时，允许发送。将待发送数据写入发送移位寄存器 0(TXS0)，可启动发送操作。起始位、校验位和停止位会自动添加到数据中。

启动发送操作时，先从 TxD0 引脚输出起始位，然后从 LSB 开始依次发送数据其它位。当发送结束时，由 ASIM0 设置的校验位和停止位被添加到数据中，并产生发送完成中断请求(INTST0)。

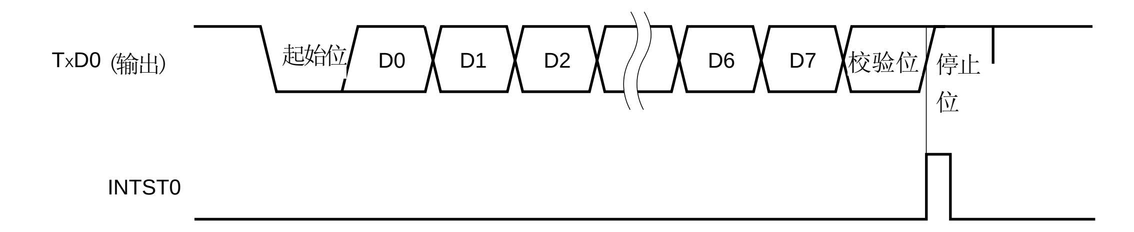
当下一个待发送数据被写入 TXS0 时，发送操作才结束。

图 14-8 显示了发送完成中断请求(INTST0)的时序。最后一个停止位一输出就产生该中断。

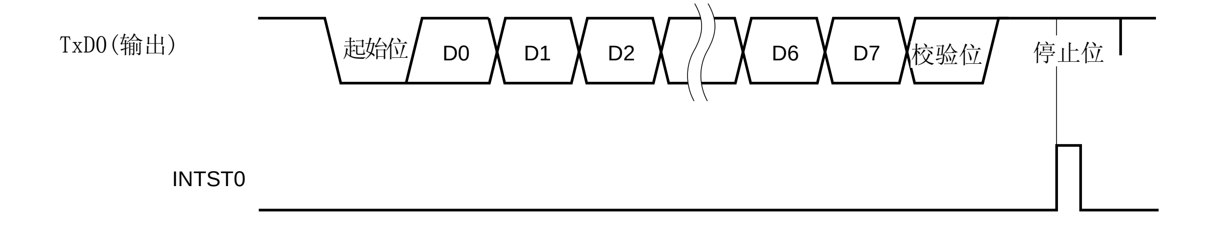
注意事项　**在将待发送数据写入 TXS0 后，在发送完成中断信号(INTST0)产生之前，不要将下一个发送数据写入 TXS0。**

图 **13-8.** 发送完成中断请求的时序

**1.** 停止位宽度**: 1**




**2.** 停止位宽度**: 2**



**(d)** 接收

当异步串行接口操作模式寄存器 0(ASIM0)的第 7 位(POWER0)=1，然后将 ASIM0 的第 5 位(RXE0)置 1 时，允许接收并对 RxD0 引脚输入进行采样。

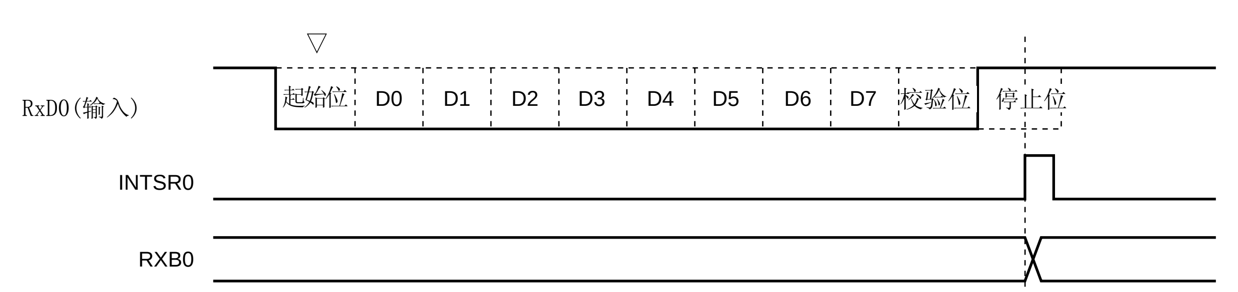
当检测到 RxD0 引脚输入的下降沿时波特率发生器的 5 位计数器开始计数。当计数值等于波特率发生器控制寄存器 0（BRGC0）的设置值时，再次采样 RxD0 的输入信号 (如图 14-9 中的 ▽)。如果此时 RxD0 为低电平，则认为输入信号为起始位。

当检测到起始位时，开始接收，以设置的波特率将串行数据依次存入接收移位寄存器 0(RXS0)。当接收到停止位时，会产生接收完成中断信号(INTSR0)，并且将 RXS0 的数据写入接收缓冲寄存器 0(RXB0)。但如果出现溢出错误，则接收到的数据不写入 RXB0。

即使在接收过程中出现校验错(PE0)，接收仍然继续，直至接收到停止位，而在接收完成后会产生一个接收错误中断(INTSR0)。

当接收操作完成且存在接收错误，则产生 INTSR0。

图 **13-9.** 接收完成中断请求时序




注意事项 **1.** 如果出现接收错误，必须先读取异步串行接口接收错误状态寄存器 **0 (ASIS0)**再读取接收缓冲寄存器 **0 (RXB0)**的内容，才能对出错标志清零。否则，当接收到下一个数据时会产生溢出错误，而且接收错误状态保持不变。

**2.** 接收始终按 “停止位的个数 **= 1**” 的情况执行，第 **2** 个停止位被忽略。

**(e)　接收错误**

在接收期间有三种出错情况: 校验错误、帧错误或溢出错误。如果异步串行接口接收错误状态寄存器 0(ASIS0)的错误标志位被设置，作为数据接收的结果，则将会产生一个接收错误中断请求信号 (INTSR0)。

在执行接收错误中断服务(INTSR0)时读取 ASIS0 的内容，可确定接收期间出现的是何种错误(参见图 **13-3**)。读取 ASIS0 后，将其清 0。

**表 13-3. 接收错误产生的原因**

| 接收错误 | 原因 |
| --- | --- |
| 校验错误 | 发送数据的校验位与接收数据的校验位不相等 |
| 帧错误 | 未检测到停止位 |
| 溢出错误 | 在从接收缓冲寄存器 0(RXB0)中读取数据前已完成下一个数据的接收操作 |

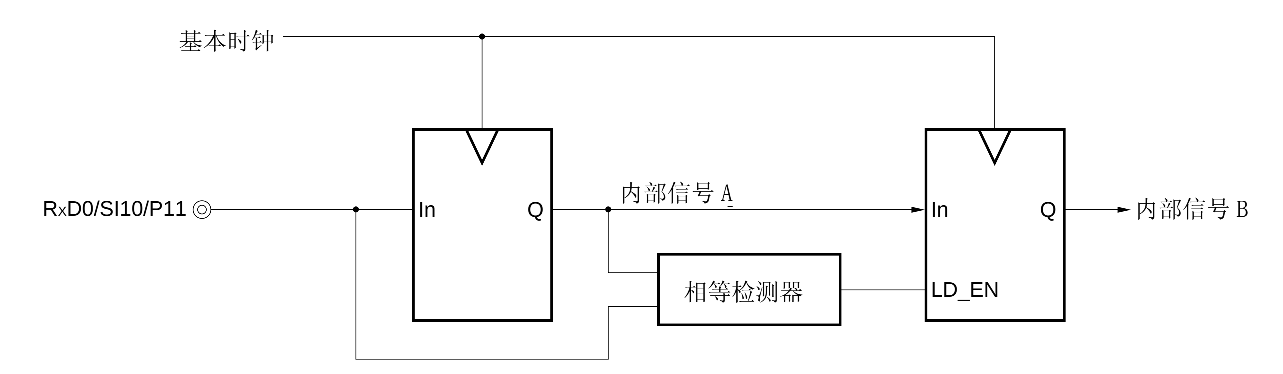
**(f)　接收数据的噪声过滤器**

通过预分频器，使用基本时钟输出，采样 RxD0 信号。

如果两次采样值相同，则相等检测器的输出会发生变化，并把采样的数据作为输入数据。

噪声过滤器的电路结构如图 13-10 所示，接收操作的内部过程与外部信号状态相比被延迟两个时钟执行。

**图 13-10. 噪声过滤器电路**



### 13.4.3 专用波特率发生器

专用波特率发生器由一个源时钟选择器和一个 5 位可编程计数器组成，用于产生 UART0 发送/接收的串行时钟。
分别使用不同的 5 位计数器用于发送和接收。

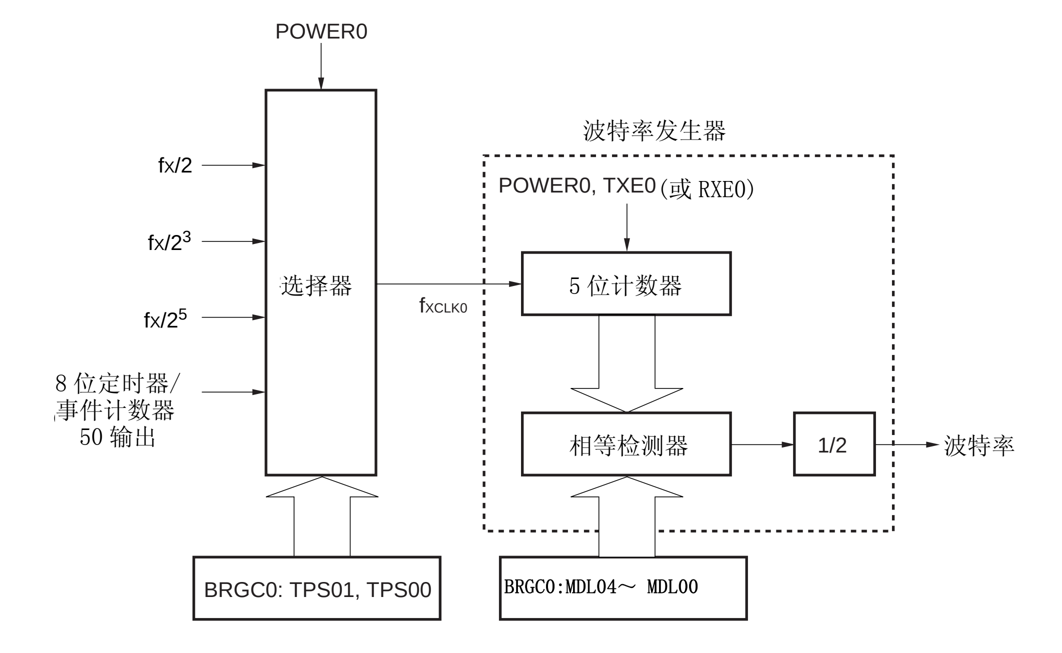
#### (1) 波特率发生器的配置

- 基本时钟
  该时钟由波特率发生器控制寄存器 0(BRGC0)的第 7 和 6 位(TPS01 和 TPS00)选择。当异步串行接口操作模式寄存器 0(ASIM0)的第 7 位(POWER0)=1 时，将该时钟提供给每个模块。该时钟称为基本时钟，它的频率用 $f_{XCLK0}$ 表示。当 POWER0 = 0 时，基本时钟恒为低电平。

- 发送计数器
  当异步串行接口操作模式寄存器 0(ASIM0)的第 7 位(POWER0)或第 6 位(TXE0) = 0 时，该计数器停止计数，并被清零。
  当 POWER0 = 1 且 TXE0 = 1 时计数器开始计数。
  当第 1 个发送数据被写入发送移位寄存器 0(TXS0)时，计数器被清零。

- 接收计数器
  当异步串行接口操作模式寄存器 0(ASIM0)的第 7 位(POWER0)或第 5 位(RXE0) = 0 时，该计数器停止计数，并被清零。
  当检测到起始位时，计数器开始计数。
  在收到一帧数据后计数器停止计数，直到下一个起始位被检测到时再重新开始计数。

**图 13-11. 波特率发生器的配置**




**备注**

| | |
|---|---|
| POWER0: | 异步串行接口操作模式寄存器 0 (ASIM0)的第 7 位 |
| TXE0: | ASIM0 的第 6 位 |
| RXE0: | ASIM0 的第 5 位 |
| BRGC0: | 波特率发生器控制寄存器 0 |

**(2) 串行时钟的产生**

使用波特率发生器控制寄存器 0 (BRGC0)可产生串行时钟。

根据 BRGC0 的第 7 和 6 位(TPS01 和 TPS00)，可选择输入给 5 位计数器的时钟。

根据 BRGC0 的 4～0 位(MDL04～MDL00)选择 5 位计数器的分频值($f_{XCLK0}/8$ ~ $f_{XCLK0}/31$)。

### 13.4.4 波特率的计算

**(1) 波特率的计算表达式**

通过下列公式计算波特率。

- 波特率= $\frac{f_{XCLK0}}{2 \times k}$ [bps]

$f_{XCLK0}$: 根据 BRGC0 的 TPS01 和 TPS00 位选择的基本时钟的频率

k: 根据 BRGC0 的 MDL04 ～ MDL00 位设置的值 (k = 8, 9, 10, ..., 31)

**表 13-4. TPS01 和 TPS00 的设置值**

| TPS01 | TPS00 | 基本时钟 ($f_{XCLK0}$) 的选择 | | | | |
|---|---|---|---|---|---|---|
| | | | $f_{PRS}$ = 2 MHz | $f_{PRS}$ = 5 MHz | $f_{PRS}$ = 10 MHz | $f_{PRS}$ = 20 MHz |
| 0 | 0 | TM50 输出 | | | | |
| 0 | 1 | $f_{PRS}/2$ | 1 MHz | 2.5 MHz | 5 MHz | 10 MHz |
| 1 | 0 | $f_{PRS}/2^3$ | 250 kHz | 625 kHz | 1.25 MHz | 2.5 MHz |
| 1 | 1 | $f_{PRS}/2^5$ | 62.5 kHz | 156.25 kHz | 312.5 kHz | 625 kHz |

**(2) 波特率误差**

通过下列公式计算波特率误差。

- 误差 (%) = $\left(\frac{\text{实际波特率（有误差的波特率）}}{\text{预期波特率（正确的波特率）}} - 1\right) \times 100$ [%]

**注意事项 1. 在发送过程中必须保持波特率误差在接收方允许的误差范围内。**

**2. 接收期间的波特率误差必须满足“(4)接收期间允许的波特率范围”中所描述的范围。**

**举例:** 基本时钟的频率 = 2.5 MHz = 2,500,000 Hz

BRGC0 的 MDL04 ～ MDL00 位 = 10000B(k = 16)

目标波特率 = 76,800 bps

波特率 = 2.5 M/(2 × 16)

= 2,500,000/(2 × 16) = 78,125 [bps]

误差 = (78,125/76,800 – 1) × 100

= 1.725 [%]

**(3)** 波特率设置举例

**表 13-5.** 波特率发生器的数据设置

| 波特率 [bps] | $f_{PRS}$ = 2.0 MHz | | | | $f_{PRS}$ = 5.0 MHz | | | | $f_{PRS}$ = 10.0 MHz | | | | $f_{PRS}$ = 20.0 MHz | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TPS01, TPS00 | k | 计算值 | ERR [%] | TPS01, TPS00 | k | 计算值 | ERR [%] | TPS01, TPS00 | k | 计算值 | ERR [%] | TPS01, TPS00 | k | 计算值 | ERR [%] |
| 4800 | 2H | 26 | 4808 | 0.16 | 3H | 16 | 4883 | 1.73 | – | – | – | – | – | – | – | – |
| 9600 | 2H | 13 | 9615 | 0.16 | 3H | 8 | 9766 | 1.73 | 3H | 16 | 9766 | 1.73 | – | – | – | – |
| 10400 | 2H | 12 | 10417 | 0.16 | 2H | 30 | 10417 | 0.16 | 3H | 15 | 10417 | 0.16 | 3H | 30 | 10417 | 0.16 |
| 19200 | 1H | 26 | 19231 | 0.16 | 2H | 16 | 19531 | 1.73 | 3H | 8 | 19531 | 1.73 | 3H | 16 | 19531 | 1.73 |
| 24000 | 1H | 21 | 23810 | –0.79 | 2H | 13 | 24038 | 0.16 | 2H | 26 | 24038 | 0.16 | 3H | 13 | 24038 | 0.16 |
| 31250 | 1H | 16 | 31250 | 0 | 2H | 10 | 31250 | 0 | 2H | 20 | 31250 | 0 | 3H | 10 | 31250 | 0 |
| 33660 | 1H | 15 | 33333 | –0.79 | 2H | 9 | 34722 | 3.34 | 2H | 18 | 34722 | 3.34 | 3H | 9 | 34722 | 3.34 |
| 38400 | 1H | 13 | 38462 | 0.16 | 2H | 8 | 39063 | 1.73 | 2H | 16 | 39063 | 1.73 | 3H | 8 | 39063 | 1.73 |
| 56000 | 1H | 9 | 55556 | –0.79 | 1H | 22 | 56818 | 1.46 | 2H | 11 | 56818 | 1.46 | 2H | 22 | 56818 | 1.46 |
| 62500 | 1H | 8 | 62500 | 0 | 1H | 20 | 62500 | 0 | 2H | 10 | 62500 | 0 | 2H | 20 | 62500 | 0 |
| 76800 | – | – | – | – | 1H | 16 | 78125 | 1.73 | 2H | 8 | 78125 | 1.73 | 2H | 16 | 78125 | 1.73 |
| 115200 | – | – | – | – | 1H | 11 | 113636 | –1.36 | 1H | 22 | 113636 | –1.36 | 2H | 11 | 113636 | –1.36 |
| 153600 | – | – | – | – | 1H | 8 | 156250 | 1.73 | 1H | 16 | 156250 | 1.73 | 2H | 8 | 156250 | 1.73 |
| <R> 312500 | – | – | – | – | – | – | – | – | 1H | 8 | 312500 | 0 | 1H | 16 | 312500 | 0 |
| <R> 625000 | – | – | – | – | – | – | – | – | – | – | – | – | 1H | 8 | 625000 | 0 |

备注　TPS01，TPS00:　波特率发生器控制寄存器 0 (BRGC0)的第 7 和 6 位 (用于设置基本时钟($f_{XCLK0}$))

k:　根据 BRGC0 的 MDL04 ～ MDL00 位设置的值 (k = 8, 9, 10, ..., 31)

$f_{PRS}$:　外围硬件时钟频率

ERR:　波特率误差

**(4)** **接收期间允许的波特率范围**

接收期间来自发送端的波特率可允许的误差范围如下所示。

**注意事项　必须确保接收期间波特率误差在允许的误差范围内，可用以下公式计算。**

**图 13-12. 接收期间允许的波特率范围**



如图 13-12 所示，当检测到起始位后，接收数据的锁存时序由波特率发生器控制寄存器 0(BRGC0)设置的计数器来确定。如果数据的最后一位(停止位)满足该锁存时序，则该数据可被正确接收。

假定接收数据为 11 位，各项理论值计算如下。

$$FL = (Brate)^{-1}$$

Brate:　UART0 的波特率

k:　BRGC0 设置的值

FL:　1 位数据宽度

锁存时序的极限值: 2 个时钟

可允许的最小数据帧宽度: $FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$

因此在发送端设定的可接收的最大波特率如下所示。

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k+2} Brate$$

同样，可计算允许的最大数据帧宽度。

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL$$

$$FLmax = \frac{21k-2}{20k} FL \times 11$$

因此在发送端设定的可接收的最小波特率如下所示。

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k-2} Brate$$

可通过上述最小和最大波特率公式计算在 UART0 与发送端之间允许的波特率误差，如下所示。

**表 13-6. 允许的最大/最小波特率误差**

| 分频比 (k) | 允许的最大波特率误差 | 允许的最小波特率误差 |
|---|---|---|
| 8 | +3.53% | –3.61% |
| 16 | +4.14% | –4.19% |
| 24 | +4.34% | –4.38% |
| 31 | +4.44% | –4.47% |

**备注**　**1.** 接收时允许的误差与每帧的位数、输入时钟频率和分频比(k)有关。输入时钟频率和分频比(k)越高，允许的误差就越大。

**2.** k: BRGC0 设置的值。

## 14.1 串行接口 UART6 的功能

串行接口 UART6 有以下两种模式。

**(1) 操作停止模式**

在不进行串行通信时可采用此模式，以降低功耗。

需要了解详细信息，可参见 **14.4.1 操作停止模式**。

**(2) 异步串行接口(UART) 模式**

该模式支持 LIN(局部互联网)-总线。该模式的功能概括如下。

需要了解详细信息，可参见 **14.4.2 异步串行接口(UART)模式** 和 **14.4.3 专用波特率发生器**。

<R>
- 最大传输率: 625 kbps
- 双引脚配置　　TxD6: 发送数据输出引脚<br>RxD6: 接收数据输入引脚
- 可选择通信数据的数据宽度为 7 或 8 位。
- 专用内置 8 位波特率发生器允许设置任意大小的波特率。
- 可分别执行发送和接收操作(全双工操作)。
- 可选 MSB-first 或 LSB-first 通信。
- 反向发送操作
- 同步中断区域发送长度为 13 ～20 位
- 可规定多于 11 位的同步中断区域接收(提供 SBF 接收标志位)。

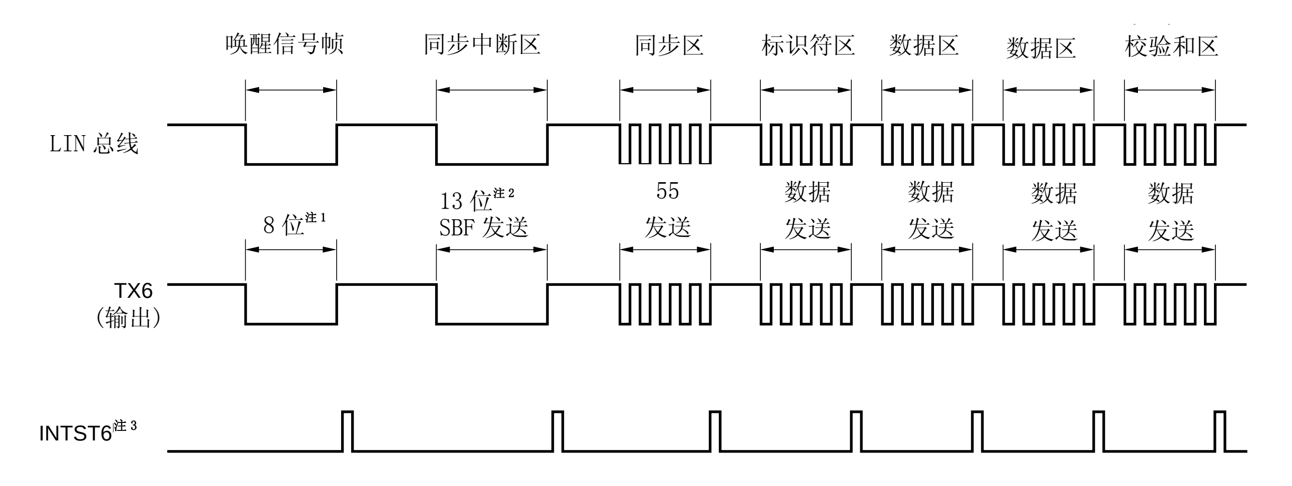
注意事项

1. **TxD6** 输出反向操作仅在发送端进行而不在接收端进行。为了能够使用该功能，接收端应准备好接收已反向的数据。
2. 如果串行接口 **UART6** 使用的时钟未被停止**(**例如，处于 **HALT** 模式下**)**，则继续正常操作。如果串行接口 **UART6** 使用的时钟已被停止**(**例如，处于 **STOP** 模式下**)**，则寄存器停止操作，并且在时钟停止之前及时保存数据。**TXD6** 引脚也在时钟停止之前立即保存数据并输出。但在时钟恢复使用后正常操作不能得到保证。因此必须复位电路，使 **POWER6 = 0**、**RXE6 = 0**、**TXE6 = 0**。
3. 设置 **POWER6 = 1** 然后设置 **TXE6 = 1 (**发送**)**或 **RXE6 = 1 (**接收**)** ，以启动通信。
4. 通过基本时钟**($f_{XCLK6}$)(**由 **CKSR6** 设置**)**对 **TXE6** 与 **RXE6** 进行同步。若要再次允许发送或接收，必须在 **TXE6** 或 **RXE6** 清零**(0)**后至少两个基本时钟时将 **TXE6** 或 **RXE6** 设置为 **1**。如果在两个基本时钟内设置 **TXE6** 或 **RXE6**，则发送电路或接收电路可能不被初始化。
5. 在设置 **TXE6 = 1** 后至少经历一个基本时钟**($f_{XCLK6}$)**再为 **TXB6** 设置发送数据。
6. 如果连续发送数据，从停止位到下一个起始位的通信时序将延长两个宏操作时钟。但这不会影响到通信结果，因为接收端会在检测到起始位时对时序初始化。如果 **UART6** 用于 **LIN** 模式，则不能使用连续发送功能。

**备注**　LIN 表示局部互联网，是一个低速(1 ～ 20 kbps)串行通信协议，可用于降低网络费用。
LIN 是一种单主设备(single－master)通信方式，一个主设备上最多可以连接 15 个从设备。
LIN 的从设备用于控制开关、制动器和传感器，这些设备通过 LIN 网与 LIN 主设备相连。
通常，LIN 主设备与一个网络，如 CAN(控制器局域网 Controller Area Network)相连。
此外，LIN 总线采用单线方式，通过收发器(符合 ISO9141)与各节点相连。
在 LIN 协议中，主设备发送带有波特率信息的一帧数据，从设备接收数据并校正波特率误差。因此当从设备端的波特率误差在 ±15%范围内时，可以进行通信。

图 14-1 和 14-2 概括了 LIN 的发送和接收操作。

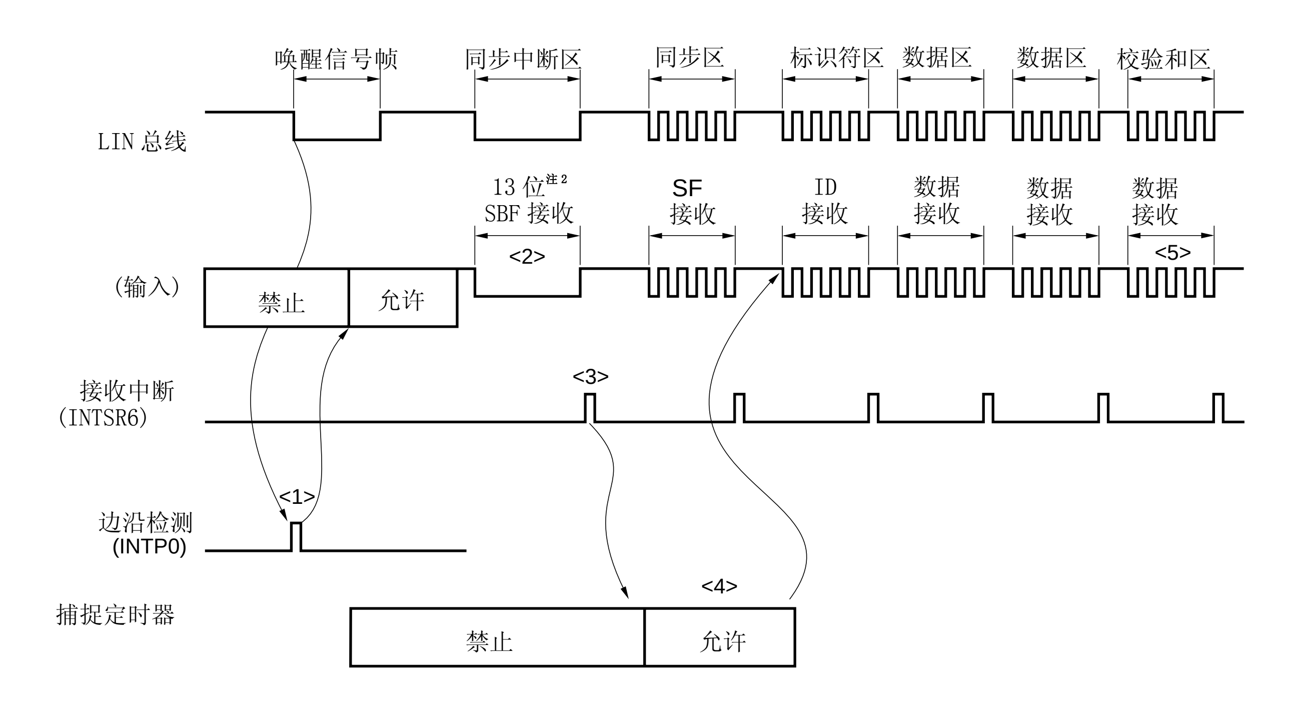
**图 14-1. LIN 发送操作**




**注**
1. 在 8 位发送模式中，可通过发送 80H 替代唤醒信号帧。
2. 使用硬件输出同步中断区。可通过异步串行接口控制寄存器 6(ASICL6)的第 4 ～2 位(SBL62 ～ SBL60)设置输出宽度(等于位长度)(参见 **14.4.2 (2) (h) SBF** 发送)。
3. 每次发送完成时输出 INTST6。在发送 SBF 时也输出 INTST6。

**备注**　区域之间的间隔由软件控制。

图 14-2. LIN 接收操作




接收处理过程如下。

<1> 在引脚的边沿处检测唤醒信号，允许使用 UART6，并设置 SBF 接收模式。

<2> 当检测到停止位时才停止接收操作。在检测到至少 11 位低电平数据的 SBF 时，认为 SBF 已被正确接收，并输出一个中断信号。在检测到少于 11 位低电平数据的 SBF 时，认为出现 SBF 接收错误。此时不输出中断信号，而是恢复 SBF 接收模式。

<3> 如果 SBF 被正确接收，将输出一个中断信号。用这个 SBF 接收完成中断服务来启动 16 位定时器/事件计数器 00 并测量同步区的位长度(脉冲宽度)(参见 **6.4.8　脉冲宽度测量操作**)。禁止检测 OVE6、PE6 和 FE6 错误，不进行 UART 通信的错误检测，也不执行移位寄存器和 RXB6 之间的数据传送。移位寄存器中保存复位值 FFH。

<4> 根据同步区的位长度计算波特率误差，在 SBF 接收后禁止使用 UART6，然后对波特率发生器控制寄存器 6(BRGC6)重新设置。

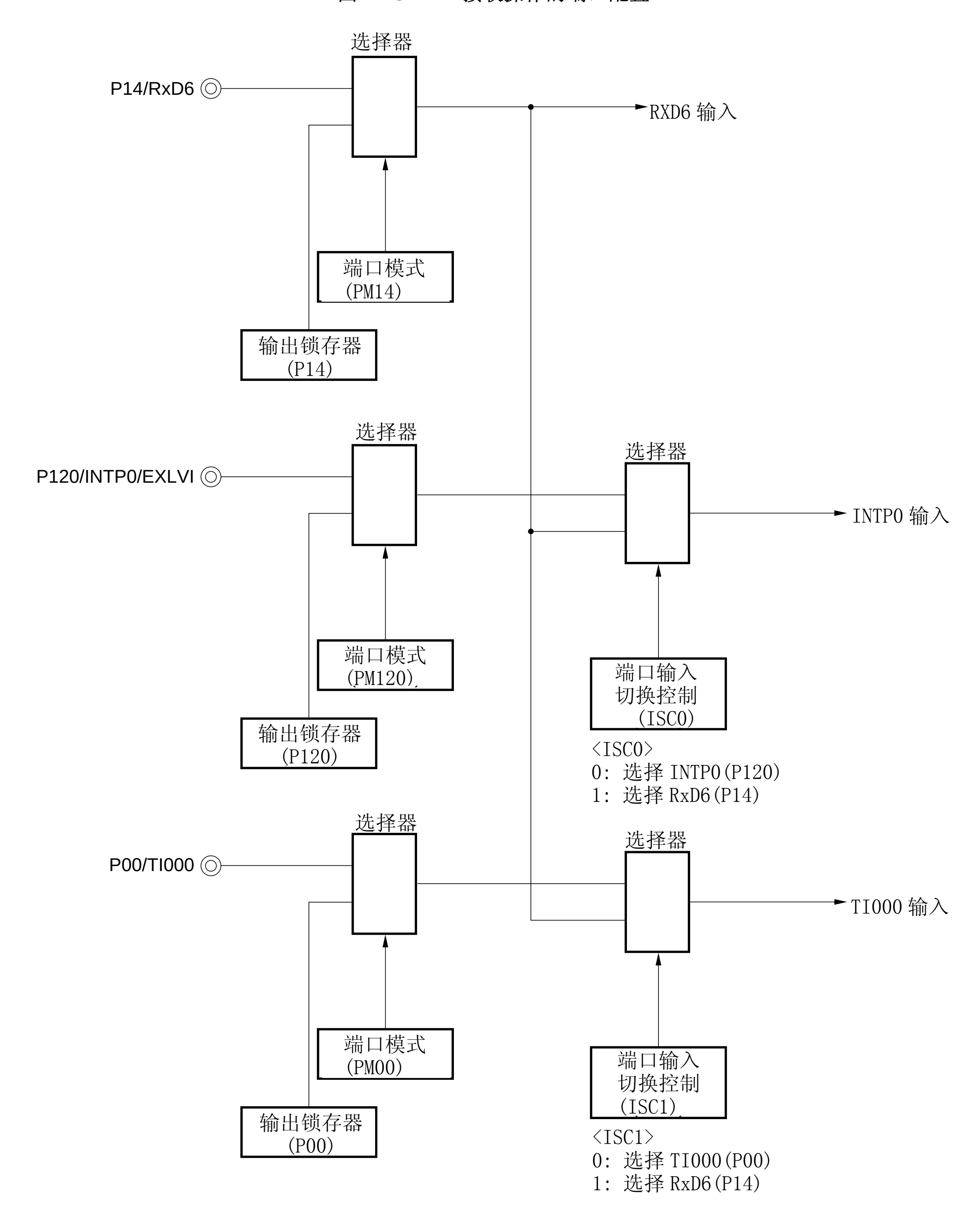
<5> 用软件区分校验和区。在接收到校验和区后通过软件对 UART6 进行初始化，并再次设置 SBF 接收模式。

图 14-3 显示了 LIN 接收操作的端口配置。

通过检测外部中断信号(INTP0)脉冲沿，接收从 LIN 主设备发出的唤醒信号。根据 16 位定时器/事件计数器 00 的外部事件捕捉操作来测量 LIN 主设备发送的同步区的位长度，并计算波特率误差。

不进行外部连接 RxD6 和 INTP0/TI000，而由端口输入切换控制(ISC0/ISC1)可将接收端口的(RxD6)的输入源输入给外部中断(INTP0)和 16 位定时器/事件计数器 00。

**图 14-3. LIN 接收操作的端口配置**




**备注**　ISC0, ISC1: 输入切换控制寄存器(ISC)的第 0 和 1 位 (参见 **图 14-11**)

以下是 LIN 通信中用到的外部功能。

<用到的外部功能>

- 外部中断(INTP0)；唤醒信号检测
  用途: 检测唤醒信号的脉冲沿和通信的开始。
- 16 位定时器/事件计数器 00(TI000)；波特率误差检测
  用途: 通过检测同步区(SF)的位长度并按位分频，来检测波特率误差(在捕捉模式下测量 TI000 输入脉冲的间隔)。
- 串行接口 UART6

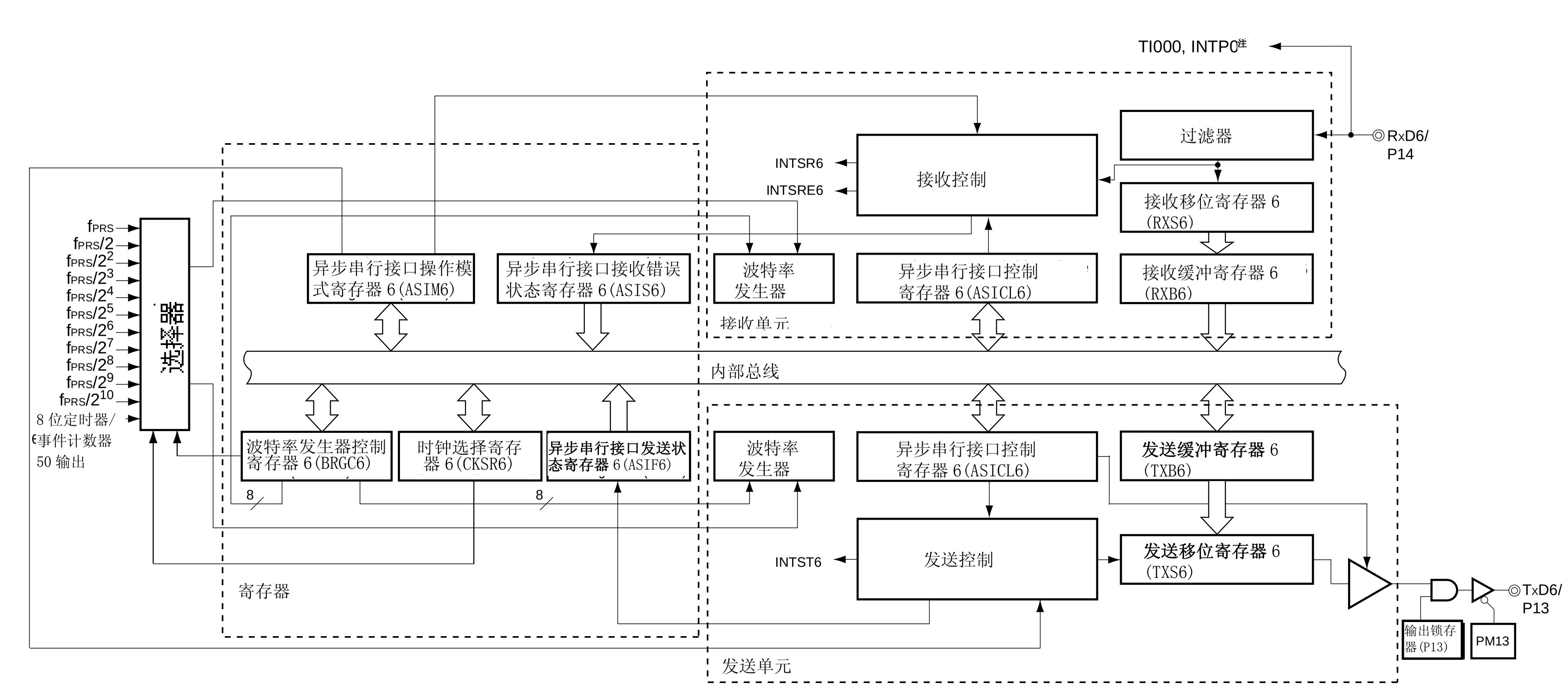
## 14.2 串行接口 UART6 的配置

串行接口 UART6 包括以下硬件。

**表 14-1. 串行接口 UART6 的配置**

| 项目 | 配置 |
|---|---|
| 寄存器 | 接收缓冲寄存器 6 (RXB6)<br>接收移位寄存器 6 (RXS6)<br>发送缓冲寄存器 6 (TXB6)<br>发送移位寄存器 6 (TXS6) |
| 控制寄存器 | 异步串行接口操作模式寄存器 6 (ASIM6)<br>异步串行接口接收错误状态寄存器 6 (ASIS6)<br>异步串行接口发送状态寄存器 6 (ASIF6)<br>时钟选择寄存器 6 (CKSR6)<br>波特率发生器控制寄存器 6 (BRGC6)<br>异步串行接口控制寄存器 6 (ASICL6)<br>输入切换控制寄存器(ISC)<br>端口模式寄存器 1 (PM1)<br>端口寄存器 1 (P1) |

图 14-4.  串行接口 UART6 的框图




注　　使用输入切换控制寄存器 (ISC) 进行选择。

**(1) 接收缓冲寄存器 6 (RXB6)**

该 8 位寄存器用于存储由接收移位寄存器(RXS6)移位的并行数据。

每接收到 1 字节的数据，新的接收数据就从 RXS6 传送到该寄存器中。如果数据宽度被设定为 7 位，则按如下方式发送数据。

- 在 LSB-first 接收模式下，将接收到的数据传送到 RXB6 的 0～6 位，而 MSB 位恒为 0。
- 在 MSB-first 接收模式下，将接收到的数据传送到 RXB6 的 1～7 位，而 LSB 位恒为 0。

如果出现溢出错误(OVE6)，则不把接收数据传送到 RXB6 中。

可由 8 位存储器操作指令读取 RXB6 的内容。不能将数据写入该寄存器中。

复位信号产生将该寄存器设置为 FFH。

**(2) 接收移位寄存器 6 (RXS6)**

该寄存器将输入到 RxD6 引脚的串行数据转换成并行数据。

程序不能直接使用 RXS6。

**(3) 发送缓冲寄存器 6 (TXB6)**

该寄存器用于设置发送数据。当数据写入 TXB6 时开始发送。

可由 8 位存储器操作指令对 TXB6 进行读或写操作。

复位信号产生将该寄存器设置为 FFH。

注意事项 **1. 当异步串行接口发送状态寄存器 6(ASIF6)的第 1 位(TXBF6)=1 时不要对 TXB6 进行写操作。**

**2. 在通信过程中(当异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)和第 6 位(TXE6)均为 1，或 ASIM6 的第 7 位(POWER6)和第 5 位(RXE6)均为 1)，不要通过软件刷新(写入相同值)TXB6。**

**3. 在设置 TXE6 = 1 后，至少经过一个基本时钟($f_{XCLK6}$)，将发送数据写入 TXB6。**

**(4) 发送移位寄存器 6 (TXS6)**

该寄存器将 TXB6 中要传送的数据作为串行数据从 TxD6 移出。在第 1 次发送过程中，当数据写入 TXB6 后立即被传送。或者在持续发送过程中，当发出 1 帧数据后，INTST6 中断产生之前数据立即被传送。利用基本时钟的下降沿从 TxD6 引脚传送来自 TXB6 的数据。

程序不能直接对 TXS6 操作。

## 14.3 控制串行接口 UART6 的寄存器

串行接口 UART6 由以下 9 个寄存器控制。

- 异步串行接口操作模式寄存器 6 (ASIM6)
- 异步串行接口接收错误状态寄存器 6 (ASIS6)
- 异步串行接口发送状态寄存器 6 (ASIF6)
- 时钟选择寄存器 6 (CKSR6)
- 波特率发生器控制寄存器 6 (BRGC6)
- 异步串行接口控制寄存器 6 (ASICL6)
- 输入切换控制寄存器(ISC)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

### (1) 异步串行接口操作模式寄存器 6 (ASIM6)

该 8 位寄存器用于控制串行接口 UART6 的串行通信。
可由 1 位或 8 位存储器操作指令设置该寄存器。
复位信号产生将该寄存器设置为 01H。

备注　在通信过程中(当异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)和第 6 位(TXE6)均为 1，或 ASIM6 的第 7 位(POWER6)和第 5 位(RXE6)均为 1)，可以通过软件刷新(写入相同值)ASIM6。

**图 14-5. 异步串行接口操作模式寄存器 6 (ASIM6)的格式 (1/2)**

地址: FF50H　复位后: 01H　R/W

| 符号 | <7> | <6> | <5> | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ASIM6 | POWER6 | TXE6 | RXE6 | PS61 | PS60 | CL6 | SL6 | ISRM6 |

| POWER6 | 允许/禁止内部操作时钟的操作 |
|---|---|
| 0[注1] | 禁止内部操作时钟的操作 (时钟恒为低电平) 并对内部电路进行异步复位[注2]。 |
| 1 | 允许内部操作时钟的操作 |

| TXE6 | 允许/禁止发送 |
|---|---|
| 0 | 禁止发送(同步复位发送电路) |
| 1 | 允许发送 |

| RXE6 | 允许/禁止接收 |
|---|---|
| 0 | 禁止接收(同步复位接收电路) |
| 1 | 允许接收 |

注　**1.** 发送期间当 POWER6 = 0 时，TxD6 引脚的输出变为高电平，而来自 RxD6 引脚的输入恒为高电平。
　　**2.** 对异步串行接口接收错误状态寄存器 6(ASIS6)、异步串行接口发送状态寄存器 6(ASIF6)、异步串行接口控制寄存器 6(ASICL6)的第 7 位(SBRF6)和第 6 位(SBRT6)，以及接收缓冲寄存器 6(RXB6)进行复位。

图 14-5. 异步串行接口操作模式寄存器 6 (ASIM6)的格式 (2/2)

| PS61 | PS60 | 发送操作 | 接收操作 |
|---|---|---|---|
| 0 | 0 | 不输出校验位 | 无校验接收 |
| 0 | 1 | 输出零校验 | 按零校验接收[注] |
| 1 | 0 | 输出奇校验 | 按奇校验判断 |
| 1 | 1 | 输出偶校验 | 按偶校验判断 |

| CL6 | 指定发送/接收数据的字符宽度 |
|---|---|
| 0 | 数据的字符宽度 = 7 位 |
| 1 | 数据的字符宽度 = 8 位 |

| SL6 | 指定发送数据停止位的个数 |
|---|---|
| 0 | 停止位的个数 = 1 |
| 1 | 停止位的个数 = 2 |

| ISRM6 | 允许/禁止出错时接收完成中断的产生 |
|---|---|
| 0 | 出错时产生“INTSRE6”(此时，不产生 INTSR6) |
| 1 | 出错时产生“INTSR6”(此时，不产生 INTSRE6) |

注　如果选择“按零校验接收”，则无法判断校验。因此不要设置异步串行接口接收错误状态寄存器 0(ASIS6)的第 2 位(PE6)，这样也不会产生出错中断。

注意事项
**1.** 启动发送操作时，对 **POWER6** 置 **1**，然后对 **TXE6** 置 **1**。停止发送操作时，先将 **TXE6** 清零，然后将 **POWER6** 清零。
**2.** 启动接收操作时，对 **POWER6** 置 **1**，然后对 **RXE6** 置 **1**。停止接收操作时，先将 **RXE6** 清零，然后将 **POWER6** 清零。
**3.** 当 **RxD6** 引脚输入为高电平时，先对 **POWER6** 置 **1**，再对 **RXE6** 置 **1**。当输入为低电平 **POWER6** 置 **1**，**RXE6** 置 **1** 时，启动接收操作。
**4.** 通过基本时钟(**$f_{XCLK6}$**)(由 **CKSR6** 设置)对 **TXE6** 与 **RXE6** 进行同步。 若要再次允许发送或接收，必须在 **TXE6** 或 **RXE6** 清零(**0**)后至少两个基本时钟时将 **TXE6** 或 **RXE6** 设置为 **1**。如果在两个基本时钟内设置 **TXE6** 或 **RXE6**，则发送电路或接收电路可能不被初始化。
**5.** 设置 **TXE6 = 1** 后，至少经过一个基本时钟(**$f_{XCLK6}$**)，将发送数据写入 **TXB6**。
**6.** 在对 **PS61**、**PS60** 和 **CL6** 位重写之前，先将 **TXE6** 和 **RXE6** 位清零。
**7.** 工作在 **LIN** 模式时，**PS61** 与 **PS60** 恒为 **0**。
**8.** 在重写 **SL6** 位时，必须确保 **TXE6 = 0**。总是以“停止位的个数=1”的格式执行接收操作，因此接收不会受到 **SL6** 设置值的影响。
**9.** 当重写 **ISRM6** 时，**RXE6** 必须为 **0**。

**(2) 异步串行接口接收错误状态寄存器 6 (ASIS6)**

该寄存器用于指示串行接口 UART6 的接收操作完成时出现的错误状态，包括 3 个出错标志位(PE6，FE6，OVE6)。

可由 8 位存储器操作指令对该寄存器进行只读操作。

如果 ASIM6 的第 5 位(RXE6)或第 7 位(POWER6) = 0，则复位信号产生将该寄存器设置为 00H。读取该寄存器的值为 00H。如果产生接收错误，读取 ASIS6 再读取接收缓冲寄存器 6 (RXB6)，可将出错标志清零。

**图 14-6. 异步串行接口接收错误状态寄存器 6 (ASIS6)的格式**

地址: FF53H 复位后: 00H R

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ASIS6 | 0 | 0 | 0 | 0 | 0 | PE6 | FE6 | OVE6 |

| PE6 | 指示校验错误的状态标志位 |
|---|---|
| 0 | 如果 POWER6 = 0 且 RXE6 = 0，或者如果读取 ASIS6 |
| 1 | 如果发送数据的校验位与接收完成时的校验位不相等 |

| FE6 | 指示帧错误的状态标志位 |
|---|---|
| 0 | 如果 POWER6 = 0 且 RXE6 = 0、或者如果读取 ASIS6 |
| 1 | 如果接收完成时未检测到停止位 |

| OVE6 | 指示溢出错误的状态标志位 |
|---|---|
| 0 | 如果 POWER6 = 0 且 RXE6 = 0、或 ASIS6 被读取 |
| 1 | 如果将接收数据存于 RXB6 寄存器中且在读取该数据前下一个接收操作已完成 |

注意事项
1. 根据异步串行接口操作模式寄存器 6(ASIM6)的 PS61 和 PS60 位的内容的不同，对 PE6 位的操作也有所不同。
2. 无论停止位的个数是多少，只将接收数据的第 1 位作为停止位进行检测。
3. 如果出现溢出错误，则下一个接收数据不写入接收缓冲寄存器 6(RXB6)中而是被忽略。
4. 如果从 ASIS6 读取数据，会产生一个等待周期。当 CPU 采用副系统时钟而停止外部硬件时钟操作时，不要从 ASIS6 读取数据。需要了解详细信息，可参见 第三十二章 等待注意事项 。

**(3) 异步串行接口发送状态寄存器 6 (ASIF6)**

该寄存器指示串行接口 UART6 的发送状态，包括两个状态标志位(TXBF6 和 TXSF6)。

当数据从 TXB6 传送到 TXS6 中后，通过将下一个发送数据写入 TXB6，使得发送能够连续进行而不被中断，即便是在中断产生期间也不受影响。

可由 8 位存储器操作指令对该寄存器进行只读操作。

如果 ASIM6 的第 7 位(POWER6)或第 6 位(TXE6)置 0，则复位信号产生将该寄存器设置为 00H。

图 **14-7.** 异步串行接口发送状态寄存器 **6 (ASIF6)**的格式

地址: FF55H 复位后: 00H R

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ASIF6 | 0 | 0 | 0 | 0 | 0 | 0 | TXBF6 | TXSF6 |

| TXBF6 | 发送缓冲数据标志位 |
|---|---|
| 0 | 如果 POWER6 = 0 或 TXE6 = 0，或如果数据被传送到发送移位寄存器 6(TXS6)中 |
| 1 | 当数据被写入发送缓冲寄存器 6(TXB6)中(如果数据已在 TXB6 中) |

| TXSF6 | 发送移位寄存器数据标志位 |
|---|---|
| 0 | 如果 POWER6 = 0 或 TXE6 = 0，或如果发送完成后下一个数据没有从发送缓冲寄存器 6(TXB6)发出 |
| 1 | 如果数据从发送缓冲寄存器 6(TXB6)发出(如果数据仍在发送过程中) |

注意事项 **1.** 为了能够连续发送数据，应将第一个发送的数据**(**第一个字节**)**写入 **TXB6** 中。之后，检查 **TXBF6** 标志位，必须为“**0**”。如果正确，将下一个发送数据**(**第二个字节**)**写入 **TXB6** 中。如果 **TXBF6** 标志为“**1**”时将数据写入 **TXB6**，则无法保证发送数据的正确性。

**2.** 如果要在连续发送结束时初始化发送单元，则在产生发送完成中断后必须检查 **TXSF6** 标志位，应为“**0**”。然后执行初始化操作。如果 **TXSF6** 标志为“**1**”时执行初始化操作，则无法保证发送数据的正确性。

**(4) 时钟选择寄存器 6 (CKSR6)**

该寄存器用于选择串行接口 UART6 的基本时钟。

可由 8 位存储器操作指令设置 CKSR6。

复位信号产生将该寄存器清零(00H)。

备注　通信过程中(当异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)和第 6 位(TXE6)均为 1，或 ASIM6 的第 7 位(POWER6)和第 5 位(RXE6)均为 1)，可由软件刷新(写入相同值)CKSR6。

**图 14-8. 时钟选择寄存器 6 (CKSR6)的格式**

地址: FF56H 复位后: 00H R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CKSR6 | 0 | 0 | 0 | 0 | TPS63 | TPS62 | TPS61 | TPS60 |

| TPS63 | TPS62 | TPS61 | TPS60 | 基本时钟($f_{XCLK6}$) 的选择 | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | $f_{PRS}$ = 2 MHz | $f_{PRS}$ = 5 MHz | $f_{PRS}$ = 10 MHz | $f_{PRS}$ = 20 MHz |
| 0 | 0 | 0 | 0 | $f_{PRS}$ | 2 MHz | 5 MHz | 10 MHz | 20 MHz |
| 0 | 0 | 0 | 1 | $f_{PRS}/2$ | 1 MHz | 2.5 MHz | 5 MHz | 10 MHz |
| 0 | 0 | 1 | 0 | $f_{PRS}/2^2$ | 500 kHz | 1.25 MHz | 2.5 MHz | 5 MHz |
| 0 | 0 | 1 | 1 | $f_{PRS}/2^3$ | 250 kHz | 625 kHz | 1.25 MHz | 2.5 MHz |
| 0 | 1 | 0 | 0 | $f_{PRS}/2^4$ | 125 kHz | 312.5 kHz | 625 kHz | 1.25 MHz |
| 0 | 1 | 0 | 1 | $f_{PRS}/2^5$ | 62.5 kHz | 156.25 kHz | 312.5 kHz | 625 kHz |
| 0 | 1 | 1 | 0 | $f_{PRS}/2^6$ | 31.25 kHz | 78.13 kHz | 156.25 kHz | 312.5 kHz |
| 0 | 1 | 1 | 1 | $f_{PRS}/2^7$ | 15.625 kHz | 39.06 kHz | 78.13 kHz | 156.25 kHz |
| 1 | 0 | 0 | 0 | $f_{PRS}/2^8$ | 7.813 kHz | 19.53 kHz | 39.06 kHz | 78.13 kHz |
| 1 | 0 | 0 | 1 | $f_{PRS}/2^9$ | 3.906 kHz | 9.77 kHz | 19.53 kHz | 39.06 kHz |
| 1 | 0 | 1 | 0 | $f_{PRS}/2^{10}$ | 1.953 kHz | 4.88 kHz | 9.77 kHz | 19.53 kHz |
| 1 | 0 | 1 | 1 | TM50 输出[注] | | | | |
| 其它情况 | | | | 禁止设置 | | | | |

注　选择 TM50 输出作为基本时钟时，应注意以下几点。

- TM50 与 CR50 相等(TMC506 = 0)时计数时钟清零并启动的模式(TMC506 = 0)
  先启动 8 位定时器/事件计数器 50，然后允许定时器 F/F 反转操作(TMC501 = 1)。
- PWM 模式 (TMC506 = 1)
  先启动 8 位定时器/事件计数器 50，然后设置计数时钟，使占空比 = 50%。

不论何种模式，都不必将 TO50 作为定时器输出引脚。

**注意事项　在重写 TPS63 ～ TPS60 位时，必须确保 POWER6 = 0。**

备注　**1.** $f_{PRS}$: 外围硬件时钟频率

**2.** TMC506: 8 位定时器模式控制寄存器 50(TMC50)的第 6 位
TMC501: TMC50 的第 1 位

**(5) 波特率发生器控制寄存器 6 (BRGC6)**

该寄存器用于设置串行接口 UART6 的 8 位计数器的分频值。

可由 8 位存储器操作指令设置 BRGC6。

复位信号产生将该寄存器设置为 FFH。

备注 通信过程中(当异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)和第 6 位(TXE6)均为 1，或 ASIM6 的第 7 位(POWER6)和第 5 位(RXE6)均为 1)，可由软件刷新(写入相同值)BRGC6。

**图 14-9. 波特率发生器控制寄存器 6 (BRGC6)的格式**

地址: FF57H 复位后: FFH R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| BRGC6 | MDL67 | MDL66 | MDL65 | MDL64 | MDL63 | MDL62 | MDL61 | MDL60 |

| | MDL67 | MDL66 | MDL65 | MDL64 | MDL63 | MDL62 | MDL61 | MDL60 | k | 8 位计数器输出时钟选择 |
|---|---|---|---|---|---|---|---|---|---|---|
| <R> | 0 | 0 | 0 | 0 | 0 | 0 | × | × | × | 禁止设置 |
| <R> | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 4 | $f_{XCLK6}/4$ |
| <R> | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 5 | $f_{XCLK6}/5$ |
| <R> | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 6 | $f_{XCLK6}/6$ |
| | • | • | • | • | • | • | • | • | • | • |
| | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 252 | $f_{XCLK6}/252$ |
| | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 253 | $f_{XCLK6}/253$ |
| | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 254 | $f_{XCLK6}/254$ |
| | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 255 | $f_{XCLK6}/255$ |

**注意事项 1. 在重写 MDL67 ～ MDL60 位时，必须确保 ASIM6 的第 6 位(TXE6)和第 5 位(RXE6) = 0。**

**2. 波特率为 8 位计数器输出时钟的 1/2。**

备注 **1.** $f_{XCLK6}$: 根据 CKSR6 寄存器的 TPS63 ～ TPS60 位选择基本时钟的频率

<R> **2.** k: 由 MDL67 ～ MDL60 位设置的值 (k = 4, 5, 6, ..., 255)

**3.** ×: 不必考虑

**(6) 异步串行接口控制寄存器 6 (ASICL6)**

该寄存器用于控制串行接口 UART6 的串行通信操作。

可由 1 位或 8 位存储器操作指令设置 ASICL6。

复位信号产生将该寄存器设置为 16H。

注意事项　**通信过程中(当异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)和第 6 位(TXE6)均为 1，或 ASIM6 的第 7 位(POWER6)和第 5 位(RXE6)均为 1)，可由软件刷新(写入相同值)ASICL6。但是，由于可能会重新触发 SBF 接收和 SBF 发送，因此在 SBF 接收期间(SBRT6 = 1)或 SBF 发送期间通过刷新操作不要将 SBRT6 与 SBTT6 都设置为 1(直到由于 SBTT6=1 而产生 INTST6 为止)。**

图 **14-10.** 异步串行接口控制寄存器 **6 (ASICL6)** 的格式**(1/2)**

地址: FF58H　复位后: 16H　R/W[注]

| 符号 | <7> | <6> | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ASICL6 | SBRF6 | SBRT6 | SBTT6 | SBL62 | SBL61 | SBL60 | DIR6 | TXDLV6 |

| SBRF6 | SBF 接收状态标志 |
|---|---|
| 0 | 如果 POWER6 = 0 且 RXE6 = 0，或者如果 SBF 已正确接收 |
| 1 | SBF 接收操作正在进行中 |

| SBRT6 | SBF 接收触发 |
|---|---|
| 0 | – |
| 1 | SBF 接收触发 |

| SBTT6 | SBF 发送触发 |
|---|---|
| 0 | – |
| 1 | SBF 发送触发 |

注　　第 7 位是只读的。

图 **15-10.**　异步串行接口控制寄存器 **6 (ASICL6)**的格式 **(2/2)**

| SBL62 | SBL61 | SBL60 | SBF 发送输出宽度控制 |
|---|---|---|---|
| 1 | 0 | 1 | SBF 输出宽度：13 位 |
| 1 | 1 | 0 | SBF 输出宽度：14 位 |
| 1 | 1 | 1 | SBF 输出宽度：15 位 |
| 0 | 0 | 0 | SBF 输出宽度：16 位 |
| 0 | 0 | 1 | SBF 输出宽度：17 位 |
| 0 | 1 | 0 | SBF 输出宽度：18 位 |
| 0 | 1 | 1 | SBF 输出宽度：19 位 |
| 1 | 0 | 0 | SBF 输出宽度：20 位 |

| DIR6 | 首位说明 |
|---|---|
| 0 | MSB |
| 1 | LSB |

| TXDLV6 | 允许/禁止反向 TxD6 输出 |
|---|---|
| 0 | TxD6 正常输出 |
| 1 | TxD6 反向输出 |

注意事项

1. 如果出现 **SBF** 接收错误，则返回到 **SBF** 接收模式，并保持 **SBRF6** 标志位状态**(1)**不变。
2. 在设置 **SBRT6** 之前，必须确保 **ASIM6** 的第 **7** 位**(POWER6)**和第 **5** 位**(RXE6)**为 **1**。在设置 **SBRT6=1** 后 **SBF** 接收完成**(**产生中断请求信号之前**)**之前不要将 **SBRT6** 清零。
3. **SBRT6** 的读取值恒为 **0**。在 **SBF** 正确接收后 **SBRT6** 自动清零。
4. 在设置 **SBTT6=1** 之前，必须确保 **ASIM6** 的第 **7** 位**(POWER6)**和第 **6** 位**(TXE6)**为 **1**。在设置 **SBTT6=1** 之后 **SBF** 接收完成**(**产生中断请求信号之前**)**之前不要将 **SBTT6** 清零。
5. **SBTT6** 的读取值恒为 **0**。并在 **SBF** 发送结束后自动清零。
6. 在接收期间不要设置 **SBRT6=1**，而在发送期间也不要设置 **SBTT6=1**。
7. 在重写 **DIR6** 和 **TXDLV6** 之前，应对 **TXE6** 和 **RXE6** 清零。

**(7) 输入切换控制寄存器(ISC)**

输入切换控制寄存器(ISC)在 LIN(Local Interconnect Network 本地互联网络)接收期间用于接收从主设备发送的状态信号。

当 ISC0 与 ISC1=1 时，选择 P14/RxD6 引脚的信号输入作为 INTP0 与 TI000 的输入源。

可由 1 位或 8 位存储器操作指令设置该寄存器。

复位信号产生将该寄存器清零(00H)。

图 **14-11. 输入切换控制寄存器(ISC)**的格式

地址: FF4FH　复位后: 00H　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ISC | 0 | 0 | 0 | 0 | 0 | 0 | ISC1 | ISC0 |

| ISC1 | TI000 输入源的选择 |
|---|---|
| 0 | TI000 (P00) |
| 1 | RxD6 (P14) |

| ISC0 | INTP0 输入源的选择 |
|---|---|
| 0 | INTP0 (P120) |
| 1 | RxD6 (P14) |

**(8) 端口模式寄存器 1 (PM1)**

该寄存器可以按位设置端口 1 的输入/输出模式。

当 P13/TxD6 引脚用于串行接口数据输出时，将 PM13 清零且 P13 的输出锁存器的值设置为 1。

当 P14/RxD6 引脚用于串行接口数据输入时，将 PM14 置 1。此时 P14 的输出锁存器的值= 0 或 1。

可由 1 位或 8 位存储器操作指令设置 PM1。

复位信号产生将该寄存器设置为 FFH。

图 **14-12.　端口模式寄存器 1 (PM1)**的格式

地址: FF21H　复位后: FFH　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM1 | PM17 | PM16 | PM15 | PM14 | PM13 | PM12 | PM11 | PM10 |

| PM1n | P1n 引脚 I/O 模式选择 (n = 0～7) |
|---|---|
| 0 | 输出模式 (输出缓冲器打开) |
| 1 | 输入模式 (输出缓冲器关闭) |

## 14.4 串行接口 UART6 的操作

串行接口 UART6 有以下两种模式。

- 操作停止模式
- 异步串行接口(UART) 模式

### 14.4.1 操作停止模式

在此模式下，不能进行串行通信，因此，可降低功耗。此外，引脚可作为一般的端口引脚使用。通过将 ASIM6 的第 7、6 和 5 位(POWER6、TXE6 和 RXE6)清零，可设置操作停止模式。

**(1)** 使用的寄存器

由异步串行接口操作模式寄存器 6(ASIM6)设置操作停止模式。

可由 1 位或 8 位存储器操作指令设置 ASIM6。

复位信号产生将该寄存器设置为 01H。

地址: FF50H 复位后: 01H R/W

| 符号 | <7> | <6> | <5> | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ASIM6 | POWER6 | TXE6 | RXE6 | PS61 | PS60 | CL6 | SL6 | ISRM6 |

| POWER6 | 允许/禁止内部操作时钟的操作 |
|---|---|
| 0[注: 1] | 禁止内部操作时钟的操作 (时钟恒为低电平) 且对内部电路进行异步复位[注 2] |

| TXE6 | 允许/禁止发送 |
|---|---|
| 0 | 禁止发送 (同步复位发送电路) |

| RXE6 | 允许/禁止接收 |
|---|---|
| 0 | 禁止接收 (同步复位接收电路) |

注 **1.** 发送期间当 POWER6 = 0 时，TxD6 的输出变成高电平而来自 RxD6 引脚的输入恒为高电平。

**2.** 对异步串行接口接收错误状态寄存器 6(ASIS6)、异步串行接口发送状态寄存器 6(ASIF6)、异步串行接口控制寄存器(ASICL6)的第 7 位(SBRF6)和第 6 位(SBRT6)和接收缓冲寄存器 6(RXB6)进行复位。

**注意事项 TXE6 和 RXE6 清零后再对 POWER6 清零，可设置操作停止模式。**
**如果要启动发送或接收操作，则先设置 POWER6=1，然后设置 TXE6=1 或 RXE6=1。**

备注 如果将 RxD6/P14 和 TxD6/P13 作为通用端口引脚使用，可参见 **第四章 端口功能** 。

### 14.4.2 异步串行接口(UART)模式

此模式下，在传送了起始位后，紧接着发送/接收 1 字节数据，然后可执行全双工操作。

可使用专用 UART 波特率发生器，这样通信选择的波特率范围较大。

**(1) 使用的寄存器**

- 异步串行接口操作模式寄存器 6 (ASIM6)
- 异步串行接口接收错误状态寄存器 6 (ASIS6)
- 异步串行接口发送状态寄存器 6 (ASIF6)
- 时钟选择寄存器 6 (CKSR6)
- 波特率发生器控制寄存器 6 (BRGC6)
- 异步串行接口控制寄存器 6 (ASICL6)
- 输入切换控制寄存器(ISC)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

在 UART 模式中设置操作的基本过程如下。

<1> 设置 CKSR6 寄存器 (参见图 **14-8**)
<2> 设置 BRGC6 寄存器 (参见图 **14-9**)
<3> 设置 ASIM6 的 0～4 位 (ISRM6、SL6、CL6、PS60、PS61) (参见 图 **14-5**)
<4> 设置 ASICL6 的第 0 和 1 位 (TXDLV6、DIR6) (参见 图 **14-10**)
<5> 设置 ASIM6 的第 7 位 (POWER6) = 1
<6> 设置 ASIM6 的第 6 位 (TXE6) = 1。 → 允许发送
设置 ASIM6 的第 5 位 (RXE6) = 1。 → 允许接收
<7> 将数据写入发送缓冲寄存器 6 (TXB6) → 开始发送数据

**注意事项　在设置端口模式寄存器和端口寄存器时，要考虑与通信另一方的关系。**

寄存器设置与引脚之间的关系如下所示。

**表 14-2. 寄存器设置与引脚之间的关系**

| POWER6 | TXE6 | RXE6 | PM13 | P13 | PM14 | P14 | UART6 操作 | 引脚功能 | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | TxD6/P13 | RxD6/P14 |
| 0 | 0 | 0 | ×[注] | ×[注] | ×[注] | ×[注] | 停止 | P13 | P14 |
| 1 | 0 | 1 | ×[注] | ×[注] | 1 | × | 接收 | P13 | RxD6 |
| | 1 | 0 | 0 | 1 | ×[注] | ×[注] | 发送 | TxD6 | P14 |
| | 1 | 1 | 0 | 1 | 1 | × | 发送/接收 | TxD6 | RxD6 |

**注**　可以用作端口寄存器。

**备注**
×:　不必考虑
POWER6: 异步串行接口操作模式寄存器 6(ASIM6)的第 7 位
TXE6:　ASIM6 的第 6 位
RXE6:　ASIM6 的第 5 位
PM1×:　端口模式寄存器
P1×:　端口输出锁存器

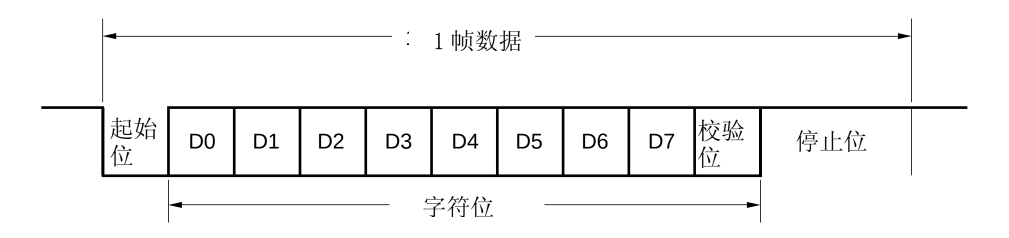
**(2)** 通信操作

**(a)** 正常发送/接收数据的格式和波形示例

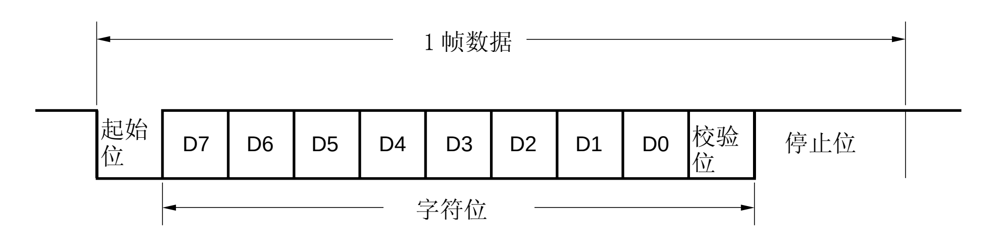
图 14-13 和 14-14 为正常发送/接收数据的格式和波形示例。

**图 14-13. 正常 UART 发送/接收数据的格式**

**1. LSB-first 发送/接收**




**2. MSB-first 发送/接收**



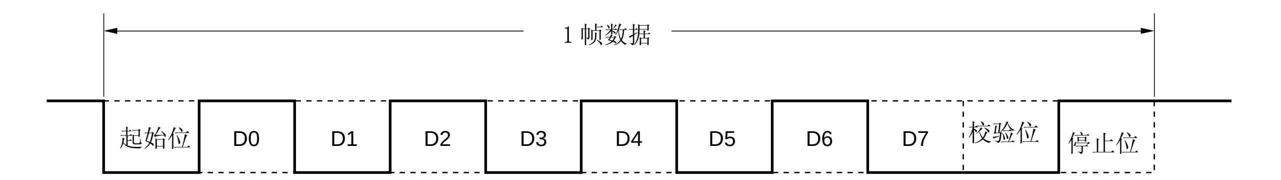

一帧数据由以下各位组成。

- 起始位 ... 1 位
- 字符位 ... 7 或 8 位
- 校验位 ... 偶校验、奇校验、零校验或无校验
- 停止位 ... 1 或 2 位

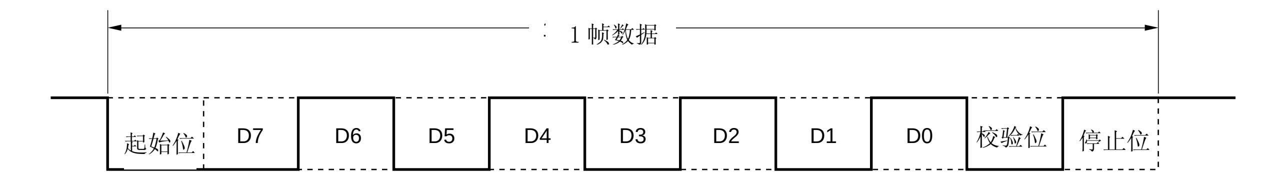
由异步串行接口操作模式寄存器 6(ASIM6)规定一个数据帧中字符位的宽度、校验方式和停止位的宽度。
由异步串行接口控制寄存器 6 (ASICL6)的第 1 位(DIR6)规定数据通信方式(LSB-first 或 MSB-first)。
由 ASICL6 的第 0 位(TXDLV6)规定 TxD6 引脚输出的是正常数据还是反向数据。

图 14-14. 正常 UART 发送/接收数据波形示例

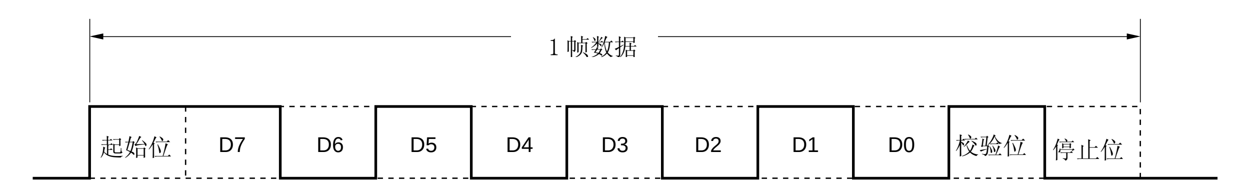
**1.** 数据宽度**: 8** 位、**LSB-first**，校验位**:** 偶校验， 停止位**: 1** 位， 通信数据**: 55H**



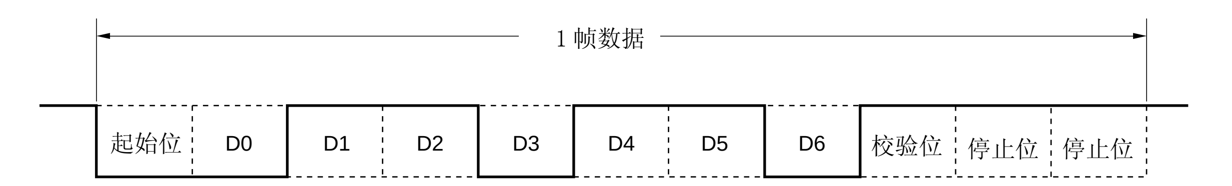

**2.** 数据宽度**: 8** 位、**MSB-first**，校验位**:** 偶校验， 停止位**: 1** 位， 通信数据**: 55H**



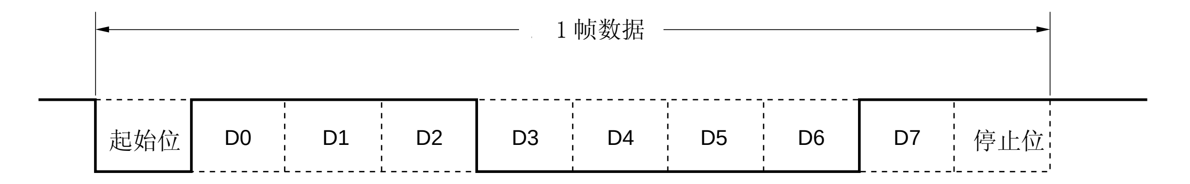

**3.** 数据宽度**: 8** 位、**MSB-first**， 校验位**:** 偶校验， 停止位**: 1** 位， 通信数据**: 55H**，**TxD6** 引脚输出反向数据




**4.** 数据宽度**: 7** 位、**LSB-first**， 校验位**:** 奇校验， 停止位**: 2** 位， 通信数据**: 36H**




**5.** 数据宽度**: 8** 位、**LSB-first**， 校验位**:** 无， 停止位**: 1** 位， 通信数据**: 87H**



**(b) 校验方式与操作**

校验位用于检测数据通信中的位错误。通常在发送和接收两端可采用相同的校验方式。当采用偶校验和奇校验时，可检测到 1 位(奇数位)错误。而采用零校验和无校验时，则不能检测到错误。

**注意事项　在 LIN 模式下，PS61 与 PS60 恒为 0。**

**(i) 偶校验**

- 发送

  控制发送数据，包括校验位，使得数据中“1”的个数为偶数。

  校验位的取值如下。

  如果发送数据有奇数个“1”: 1

  如果发送数据有偶数个“1”: 0

- 接收

  计算接收数据(包括校验位)中 “1”的个数。如果“1”的个数为奇数，则产生校验错。

**(ii) 奇校验**

- 发送

  控制发送数据，包括校验位，使得数据中“1”的个数为奇数。

  如果发送数据有奇数个“1”: 0

  如果发送数据有偶数个“1”: 1

- 接收

  计算接收数据(包括校验位)中“1”的个数。如果“1”的个数为偶数，则产生校验错。

**(iii) 零校验**

无论发送数据为何值，发送时校验位清零。

接收数据时不检测校验位。因此无论校验位为何值，都不会产生校验错。

**(iv) 无校验**

发送数据中没有校验位。

进行接收操作时认为接收数据中没有校验位。因此也不会产生校验错。

**(c)** 正常发送

在设置异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)=1 然后设置 ASIM6 的第 6 位(TXE6)=1 时，则允许发送。将待发送的数据写入发送缓冲寄存器 6(TXB6)中，可启动发送操作。起始位、校验位和停止位会自动被添加到数据中。

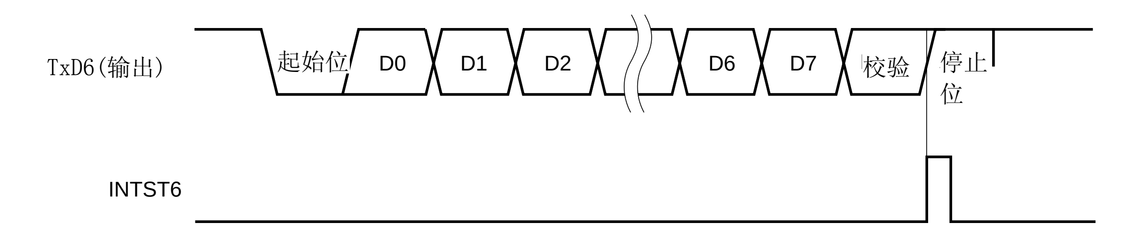
启动发送操作时，TXB6 中的数据被传送到发送移位寄存器 6(TXS6)中，然后数据依次从 TXS6 被输出到 TxD6 引脚。当发送结束时，由 ASIM6 设置的校验位和停止位被添加到数据中，并产生发送完成中断请求(INTST6)。

当下一个发送数据写入 TXB6 时，发送操作才结束。

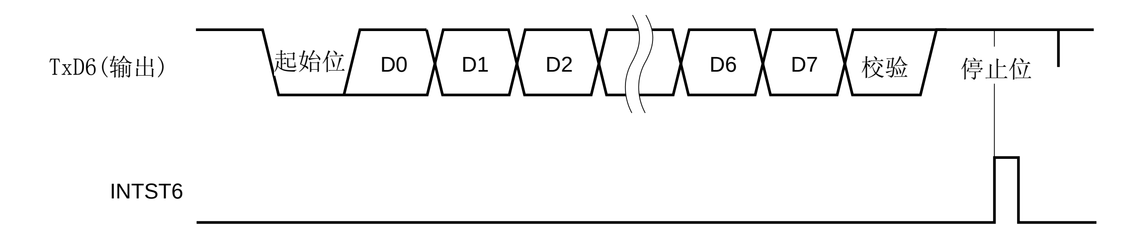
图 15-15 显示了发送完成中断请求(INTST6)的时序。最后一个停止位一输出就产生该中断。

图 **14-15.** 正常的发送完成中断请求时序

**1.** 停止位宽度**: 1**




**2.** 停止位宽度**: 2**



**(d)** 连续发送

当发送移位寄存器 6 (TXS6)一启动移位操作，就可以把下一个发送数据写入发送缓冲寄存器 6(TXB6)。因此，当发送完一帧数据后，即使在执行 INTST6 中断服务时也能够连续发送数据，从而实现高效率的通信速率。此外，当产生发送完成中断时通过读取 ASIF6 的第 0 位(TXSF6)，可以对 TXB6 进行两次有效的写操作(2 个字节)而无须等待发送一帧数据的时间。

为了能够连续发送数据，必须根据 ASIF6 检查发送状态，判断是否可以对 TXB6 进行写操作，如果可以，则写入数据。

注意事项 **1.** **在连续发送期间，ASIF6 寄存器的 TXBF6 和 TXSF6 标志从“10” 变成 “11”， 再变成 “01”。因此检查状态时，不能通过 TXBF6 和 TXST6 来判断。在连续发送期间只读 TXBF6 标志。**

**2.** **在 LIN 模式下，不能使用连续发送功能。在将发送数据写入发送缓冲寄存器 6(TXB6)之前，异步串行接口发送状态寄存器 6(ASIF6)的值必须为 00H。**

| TXBF6 | 对 TXB6 的写操作 |
|---|---|
| 0 | 允许写操作 |
| 1 | 禁止写操作 |

注意事项 **为了能够连续发送数据，应将第一个发送的数据(第一个字节)写入 TXB6 中。之后，检查 TXBF6 标志位，必须为“0”。如果正确，将下一个发送数据(第二个字节)写入 TXB6 中。如果 TXBF6 标志为“1”时将数据写入 TXB6，则无法保证发送数据的正确性。**

可用 TXSF6 标志位检测通信状态。

| TXSF6 | 发送状态 |
|---|---|
| 0 | 发送操作结束 |
| 1 | 发送操作正在进行中 |

注意事项 **1.** **如果要在连续发送结束时初始化发送单元，则在产生发送完成中断后必须检查 TXSF6 标志位，应为“0”。然后执行初始化操作。如果 TXSF6 标志为“1”时执行初始化操作，则无法保证发送数据的正确性。**

**2.** **在连续发送期间，当发送完一帧数据后执行 INTST6 中断服务之前下一个发送操作可能已经完成。计数测量时，可通过开发一个可以算发送数据的个数的程序或者通过参考 TXSF6 标志来实现检测。**

图 14-16 为连续发送操作流程示例。

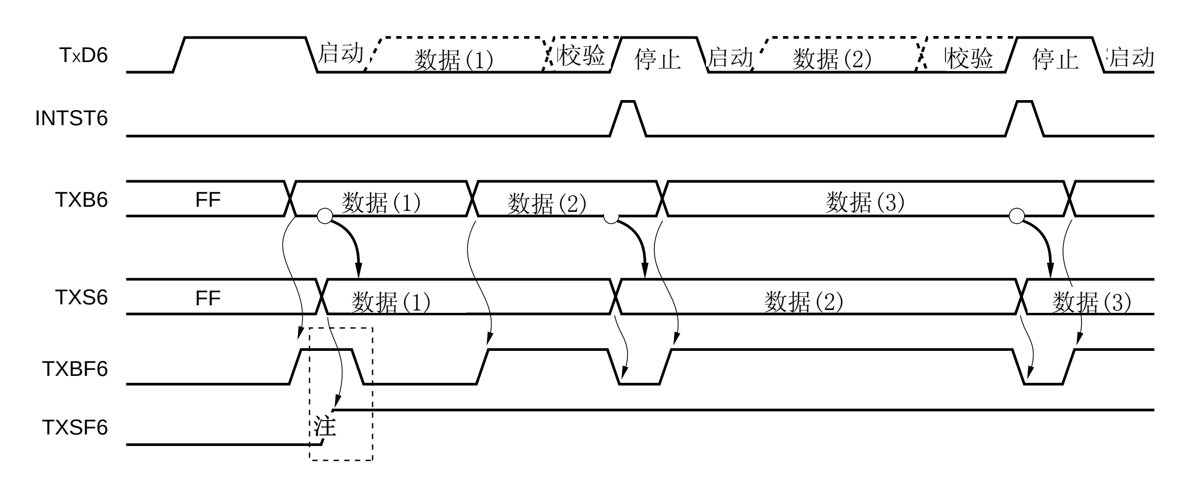
**图 14-16. 连续发送操作流程示例**



**备注**　TXB6:　发送缓冲寄存器 6
ASIF6:　异步串行接口发送状态寄存器 6
TXBF6:　ASIF6 的第 1 位(发送缓冲器数据标志位)
TXSF6:　ASIF6 的第 0 位(发送移位寄存器数据标志位)

图 14-17 显示了启动连续发送操作的时序，而图 14-18 显示了结束连续发送操作的时序。

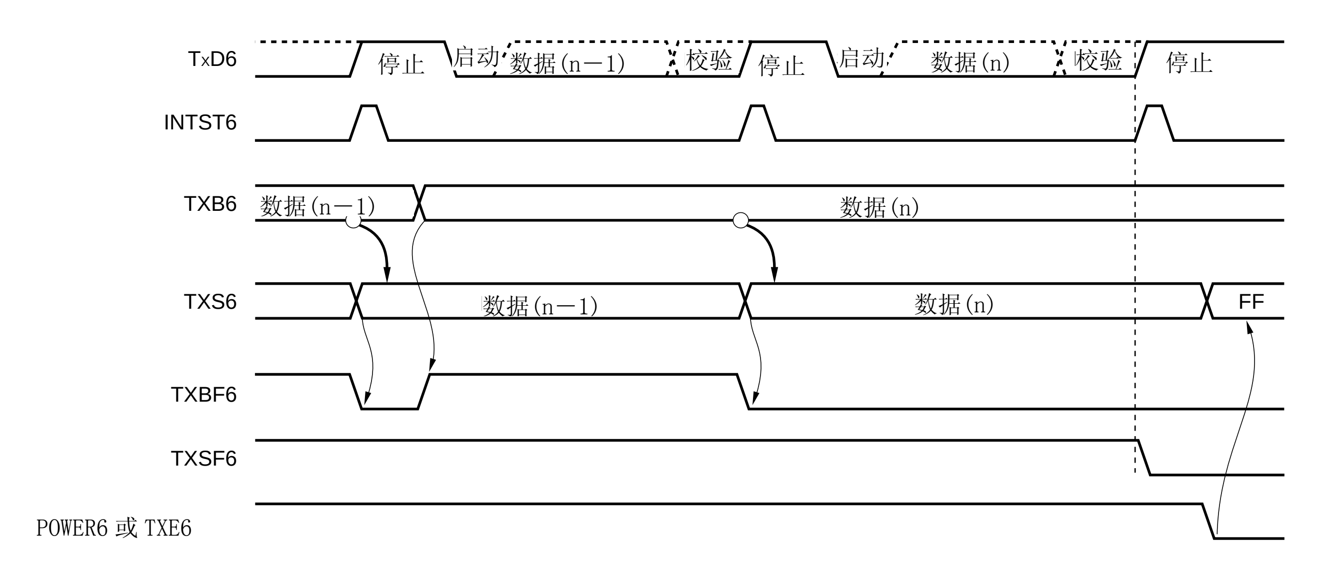
**图 14-17. 启动连续发送操作的时序**




注　在读取 ASIF6 时，存在 TXBF6 和 TXSF6 = 1，1 的情况。因此，仅用 TXBF6 就可以判断是否能进行写操作。

备注　TxD6:　TxD6 引脚 (输出)
INTST6:　中断请求信号
TXB6:　发送缓冲寄存器 6
TXS6:　发送移位寄存器 6
ASIF6:　异步串行接口发送状态寄存器 6
TXBF6:　ASIF6 的第 1 位
TXSF6:　ASIF6 的第 0 位

图 14-18. 结束连续发送操作的时序




| 备注 | | |
|---|---|---|
| | TxD6: | TxD6 引脚 (输出) |
| | INTST6: | 中断请求信号 |
| | TXB6: | 发送缓冲寄存器 6 |
| | TXS6: | 发送移位寄存器 6 |
| | ASIF6: | 异步串行接口发送状态寄存器 6 |
| | TXBF6: | ASIF6 的第 1 位 |
| | TXSF6: | ASIF6 第 0 位 |
| | POWER6: | 异步串行接口操作模式寄存器(ASIM6)的第 7 位 |
| | TXE6: | 异步串行接口操作模式寄存器(ASIM6)的第 6 位 |

**(e)** 正常接收

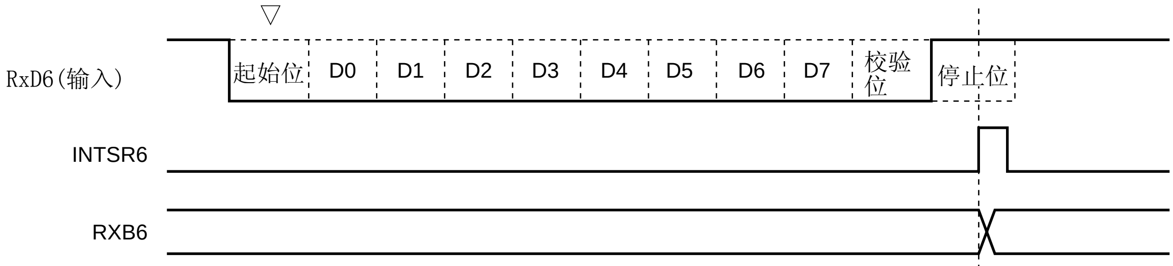
当设置异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)=1，然后将 ASIM0 的第 5 位(RXE0)置 1 时，允许接收并对 RxD6 引脚的输入进行采样。

当检测到 RxD6 引脚输入的下降沿时，波特率发生器的 8 位计数器开始计数。当计数值达到波特率发生器控制寄存器 6(BRGC6)的设置值时，再次采样 RxD6 的输入信号(如图 14-19 中的 ▽)。如果此时 RxD6 为低电平，则认为输入信号为起始位。

当检测到起始位时，开始接收，以设置的波特率将串行数据依次存入接收移位寄存器 6(RXS6)中。当接收到停止位时，会产生接收完成中断信号(INTSR6)，并且 RXS6 的数据被写入接收缓冲寄存器 6(RXB6)中。但如果出现溢出错误(OVE6)，则接收到的数据不写入 RXB6。

即使在接收过程中出现校验错(PE6)，接收仍然继续，直至接收到停止位，而在接收完成后会产生一个接收错误中断(INTSR6/INTSRE6)。

**图 14-19. 接收完成中断请求时序**




注意事项 **1.** 如果出现接收错误，则先后读取 **ASIS6** 和 **RXB6**，可将出错标志清零。否则，当接收到下一个数据时会产生溢出错误，而且接收错误状态保持不变。

**2.** 接收始终按“停止位的个数 = 1”的情况执行，第 **2** 个停止位被忽略。

**3.** 在读取 **RXB6** 之前，必须先读取异步串行接口接收错误状态寄存器 **6(ASIS6)**。

**(f)** 接收错误

在接收期间有三种出错情况: 校验错误、帧错误或溢出错误。作为数据接收的结果，作为数据接收的结果，如果异步串行接口接收错误状态寄存器 6(ASIS6)的出错标志位被置位，则将会产生一个接收错误中断请求信号(INTSR6/INTSRE6)。

在执行接收错误中断服务(INTSR6/INTSRE6)时读取 ASIS6 的内容，可确定接收期间出现的是何种错误(参见**图 14-6**)。

读取 ASIS6 后，其内容清零(0)。

表 **14-3.** 产生接收错误的原因

| 接收错误 | 产生原因 |
| --- | --- |
| 校验错误 | 发送数据的校验位与接收到的数据校验位不相等 |
| 帧错误 | 未检测到停止位 |
| 溢出错误 | 在读取接收缓冲寄存器 6(RXB6)的数据前已完成下一个数据的接收操作 |

通过将异步串行接口操作模式寄存器 6(ASIM6)的第 0 位(ISRM6)清零，可以把接收错误中断划分为接收完成中断(INTSR6)和错误中断(INTSRE6)两部分。

图 **14-20.** 接收错误中断

**1.** 如果 **ISRM6** 被清零**(**接收完成中断 **(INTSR6)** 和接收错误中断 **(INTSRE6)**是分开的**)**

**(a)** 接收期间无错误

INTSR6

INTSRE6

**(b)** 接收期间产生错误

INTSR6

INTSRE6

**2.** 如果 **ISRM6=1 (**接收错误中断包含在 **INTSR6** 中**)**

**(a)** 接收期间无错误

INTSR6

INTSRE6

**(b)** 接收期间产生错误

INTSR6

INTSRE6

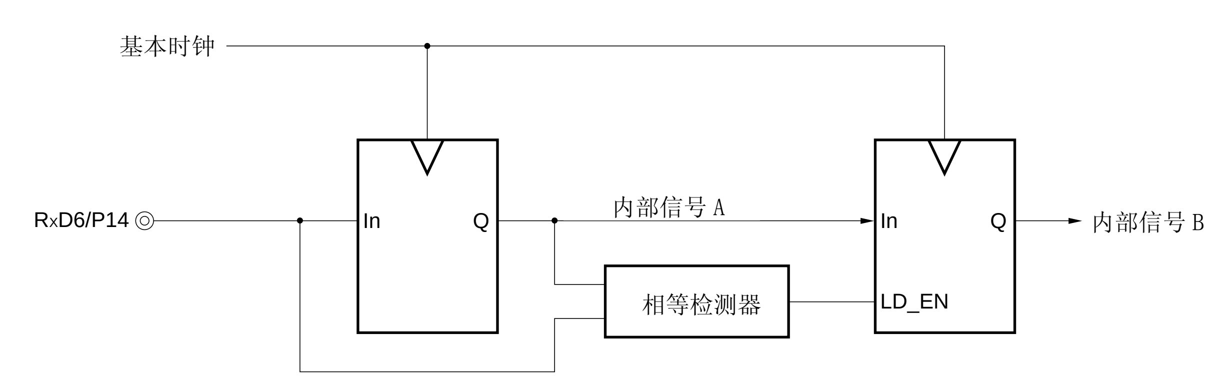
**(g)　接收数据的噪声过滤器**

使用由预分频模块输出的基准时钟采样 RXD6 信号。

如果两次采样值相同，则相等检测器的输出会发生变化，并把采样的数据作为输入数据。

噪声过滤器的电路结构如图 15-21 所示，接收操作的内部过程从外部信号状态开始延迟两个时钟才执行。

**图 14-21.　噪声过滤器电路**



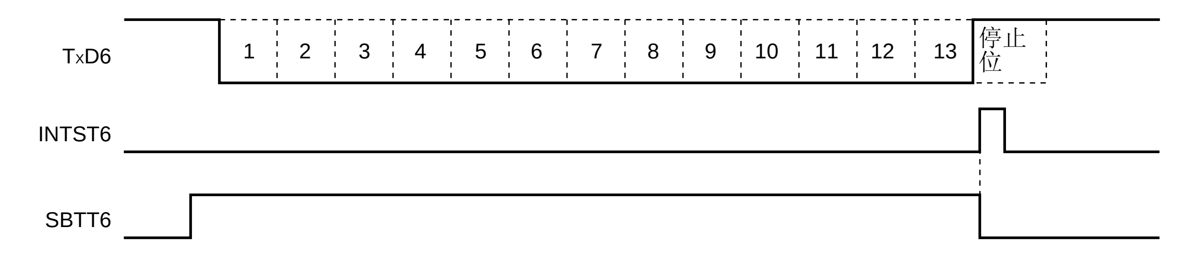

**(h)　SBF 发送**

在 LIN 模式中，可使用 SBF (同步中断区)发送控制功能进行发送。LIN 的发送操作参见**图 14-1　LIN 的发送操作**。

当异步串行接口模式寄存器 6(ASIM6)的第 7 位(POWER6)被设置为 1 时，TxD6 引脚输出高电平。接下来 ASIM6 的第 6 位(TXE6)被设置为 1 时，进入允许发送状态。且通过设置异步串行接口控制寄存器 6(ASICL6)的第 5 位(SBTT6)为 1，可启动 SBF 发送操作。

开始发送后，第 13～20 位(由 ASICL6 的第 4～2 位(SBL62 ～ SBL60)设置)输出低电平。SBF 发送完毕后，产生发送完成中断请求(INTST6)，且 SBTT6 被自动清零。接着，恢复到正常发送模式。直到下一次数据被发送到缓冲寄存器 6(TXB6)或 SBTT6 被设置为 1 时，才停止 SBF 发送。

**图 14-22.　SBF 发送**




**备注**　TxD6:　TxD6 引脚 (输出)

INTST6:　发送完成中断请求

SBTT6:　异步串行接口控制寄存器 6 (ASICL6) 的第 5 位

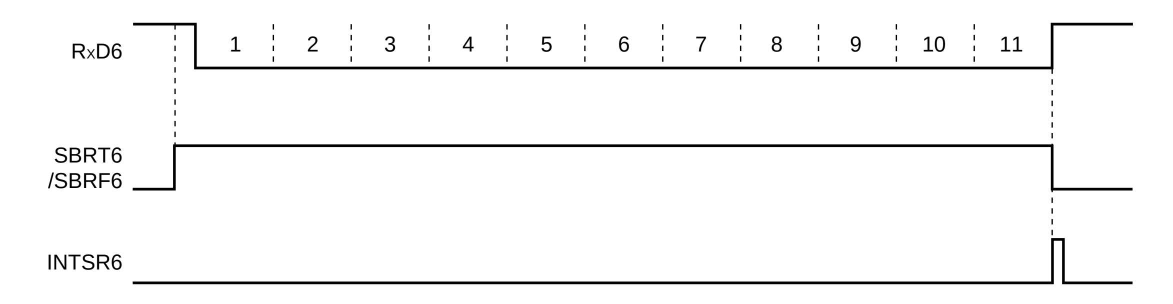
**(i)　SBF 接收**

在 LIN 模式下，SBF 接收控制功能用于接收操作。需要了解 LIN 的接收操作，可参见**图 14-2　LIN 接收操作**。将 ASIM6 的第 7 位(POWER6)被设置为 1，然后再将 ASIM6 的第 5 位(RXE6)设置为 1，允许接收。当异步串行接口控制寄存器 6(ASICL6)的第 6 位(SBRT6)被设置为 1 时，允许 SBF 接收。在 SBF 接收允许状态下，对 RxD6 引脚进行采样，并以与正常接收允许状态下同样的方式检测起始位。

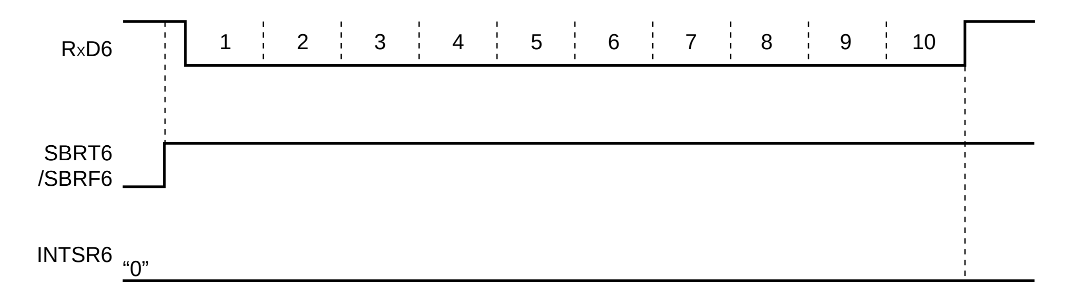
当检测到起始位时，启动接收操作，按照设置的波特率将串行数据依次存入接收移位寄存器 6(RXS6)中。在接收到停止位且 SBF 的宽度大于或等于 11 位时，与正常接收操作相同，将产生接收完成中断请求(INTSR6)。此时 SBRF6 和 SBRT6 被自动清零，SBF 接收操作结束。禁止检测 OVE6、PE6 和 FE6(异步串行接口接收错误状态寄存器 6(ASIS6)的 0 ～ 2 位)错误，不进行 UART 通信的错误检测。此外不执行接收移位寄存器 6(RXS6)和接收缓冲寄存器 6(RXB6)之间的数据传送，并保存复位值 FFH。如果 SBF 的宽度小于或等于 10 位，则在接收到停止位后不会产生出错中断，并恢复 SBF 接收模式。此时不对 SBRF6 和 SBRT6 位清零。

**图 14-23.　SBF 接收**

**1.　正常 SBF 接收 (检测到停止位，且 SBF 宽度大于 10.5 位)**




**2.　SBF 接收错误 (检测到停止位，且 SBF 宽度小于或等于 10.5 位)**




备注　RxD6:　RxD6 引脚 (输入)
SBRT6:　异步串行接口控制寄存器 6 (ASICL6) 的第 6 位
SBRF6:　ASICL6 的第 7 位
INTSR6:　接收完成中断请求

### 14.4.3 专用波特率发生器

专用波特率发生器由一个源时钟选择器和一个 8 位可编程计数器组成，用于产生 UART6 发送/接收的串行时钟。
分别使用不同的 8 位计数器用于发送和接收。

**(1) 波特率发生器的配置**

- 基本时钟
  该时钟由时钟选择寄存器 6(CKSR6)的第 3～0 位(TPS63～TPS60)选择。当异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)=1 时，将该时钟提供给每个模块。该时钟称为基本时钟，它的频率用 $f_{XCLK6}$ 表示。当 POWER6 = 0 时，基本时钟恒为低电平。

- 发送计数器
  当异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)或第 6 位(TXE6) = 0 时，该计数器停止计数，并被清零。
  当 POWER6 = 1 且 TXE6 = 1 时计数器开始计数。
  当第 1 个发送数据写入发送缓冲寄存器 6(TXB6)时，计数器被清零。
  如果连续发送数据，则当发送完 1 帧数据时计数器再次被清零。如果后面没有数据要发送，则计数器不被清零，而是继续计数直至 POWER6 或 TXE6 = 0。

- 接收计数器
  当异步串行接口操作模式寄存器 6(ASIM6)的第 7 位(POWER6)或第 5 位(RXE6)= 0 时，该计数器停止计数，并被清零。
  当检测到起始位时，计数器开始计数。
  在收到一帧数据后计数器停止计数，直到下一个起始位被检测到时再重新开始计数。

图 **14-24.** 波特率发生器的配置



**备注**　POWER6:　异步串行接口操作模式寄存器 6 (ASIM6)的第 7 位
TXE6:　ASIM6 的第 6 位
RXE6:　ASIM6 的第 5 位
CKSR6:　时钟选择寄存器 6
BRGC6:　波特率发生器控制寄存器 6

**(2)** 串行时钟的产生

使用时钟选择寄存器 6(CKSR6)和波特率发生器控制寄存器 6 (BRGC6)可产生串行时钟。

使用 CKSR6 的第 3 ～ 0 位(TPS63 ～ TPS60)，可以设置输入到 8 位计数器的时钟。使用 BRGC6 的 7 ～ 0 位

<R>　( MDL67 ～ MDL60)设置 8 位计数器的分频值($f_{XCLK6}/4$ ～ $f_{XCLK6}/255$)。

### 14.4.4 波特率的计算

**(1)** 波特率计算表达式

通过下列公式计算波特率。

- 波特率 $= \frac{f_{XCLK6}}{2 \times k}$ [bps]

$f_{XCLK6}$: 根据 CKSR6 的 TPS63 ～ TPS60 位选择的基本时钟的频率

k: 根据 BRGC6 的 MDL67 ～ MDL60 位设置的值(k = 4, 5, 6, ..., 255)

**表 14-4. TPS63 ～ TPS60 的设置值**

| TPS63 | TPS62 | TPS61 | TPS60 | 基本时钟 ($f_{XCLK6}$) 的选择 | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | $f_{PRS}$ = 2 MHz | $f_{PRS}$ = 5 MHz | $f_{PRS}$ = 10 MHz | $f_{PRS}$ = 20 MHz |
| 0 | 0 | 0 | 0 | $f_{PRS}$ | 2 MHz | 5 MHz | 10 MHz | 20 MHz |
| 0 | 0 | 0 | 1 | $f_{PRS}/2$ | 1 MHz | 2.5 MHz | 5 MHz | 10 MHz |
| 0 | 0 | 1 | 0 | $f_{PRS}/2^2$ | 500 kHz | 1.25 MHz | 2.5 MHz | 5 MHz |
| 0 | 0 | 1 | 1 | $f_{PRS}/2^3$ | 250 kHz | 625 kHz | 1.25 MHz | 2.5 MHz |
| 0 | 1 | 0 | 0 | $f_{PRS}/2^4$ | 125 kHz | 312.5 kHz | 625 kHz | 1.25 MHz |
| 0 | 1 | 0 | 1 | $f_{PRS}/2^5$ | 62.5 kHz | 156.25 kHz | 312.5 kHz | 625 kHz |
| 0 | 1 | 1 | 0 | $f_{PRS}/2^6$ | 31.25 kHz | 78.13 kHz | 156.25 kHz | 312.5 kHz |
| 0 | 1 | 1 | 1 | $f_{PRS}/2^7$ | 15.625 kHz | 39.06 kHz | 78.13 kHz | 156.25 kHz |
| 1 | 0 | 0 | 0 | $f_{PRS}/2^8$ | 7.813 kHz | 19.53 kHz | 39.06 kHz | 78.13 kHz |
| 1 | 0 | 0 | 1 | $f_{PRS}/2^9$ | 3.906 kHz | 9.77 kHz | 19.53 kHz | 39.06 kHz |
| 1 | 0 | 1 | 0 | $f_{PRS}/2^{10}$ | 1.953 kHz | 4.88 kHz | 9.77 kHz | 19.53 kHz |
| 1 | 0 | 1 | 1 | TM50 输出 | | | | |
| 其它情况 | | | | 禁止设置 | | | | |

**(2)** 波特率误差

通过下列公式计算波特率误差。

- 误差 (%) = $\left( \frac{\text{实际波特率（有误差的波特率）}}{\text{预期波特率（正确的波特率）}} - 1 \right) \times 100$ [%]

**注意事项 1. 发送期间必须保持波特率误差在接收端允许的误差范围内。**

**2. 接收期间，波特率误差必须满足"(4)接收期间允许的波特率范围"中所描述的范围。**

**举例:** 基本时钟频率 = 10 MHz = 10，000，000 Hz
BRGC6 的 MDL67 ～ MDL60 位 = 00100001B (k = 33)
目标波特率 = 153600 bps

波特率 = 10 M / (2 × 33)
= 10000000 / (2 × 33) = 151,515 [bps]

误差 = (151515/153600 − 1) × 100
= −1.357 [%]

### **(3)** 波特率设置举例

**表 14-5.** 波特率发生器的设置数据

| 波特率 [bps] | $f_{PRS}$ = 2.0 MHz | | | | $f_{PRS}$ = 5.0 MHz | | | | $f_{PRS}$ = 10.0 MHz | | | | $f_{PRS}$ = 20.0 MHz | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TPS63-TPS60 | k | 计算值 | ERR [%] | TPS63-TPS60 | k | 计算值 | ERR [%] | TPS63-TPS60 | k | 计算值 | ERR [%] | TPS63-TPS60 | k | 计算值 | ERR [%] |
| 300 | 8H | 13 | 301 | 0.16 | 7H | 65 | 301 | 0.16 | 8H | 65 | 301 | 0.16 | 9H | 65 | 301 | 0.16 |
| 600 | 7H | 13 | 601 | 0.16 | 6H | 65 | 601 | 0.16 | 7H | 65 | 601 | 0.16 | 8H | 65 | 601 | 0.16 |
| 1200 | 6H | 13 | 1202 | 0.16 | 5H | 65 | 1202 | 0.16 | 6H | 65 | 1202 | 0.16 | 7H | 65 | 1202 | 0.16 |
| 2400 | 5H | 13 | 2404 | 0.16 | 4H | 65 | 2404 | 0.16 | 5H | 65 | 2404 | 0.16 | 6H | 65 | 2404 | 0.16 |
| 4800 | 4H | 13 | 4808 | 0.16 | 3H | 65 | 4808 | 0.16 | 4H | 65 | 4808 | 0.16 | 5H | 65 | 4808 | 0.16 |
| 9600 | 3H | 13 | 9615 | 0.16 | 2H | 65 | 9615 | 0.16 | 3H | 65 | 9615 | 0.16 | 4H | 65 | 9615 | 0.16 |
| 19200 | 2H | 13 | 19231 | 0.16 | 1H | 65 | 19231 | 0.16 | 2H | 65 | 19231 | 0.16 | 3H | 65 | 19231 | 0.16 |
| 24000 | 1H | 21 | 23810 | –0.79 | 3H | 13 | 24038 | 0.16 | 4H | 13 | 24038 | 0.16 | 5H | 13 | 24038 | 0.16 |
| 31250 | 1H | 4 | 31250 | 0 | 4H | 5 | 31250 | 0 | 5H | 5 | 31250 | 0 | 6H | 5 | 31250 | 0 |
| 38400 | 1H | 13 | 38462 | 0.16 | 0H | 65 | 38462 | 0.16 | 1H | 65 | 38462 | 0.16 | 2H | 65 | 38462 | 0.16 |
| 48000 | 0H | 21 | 47619 | –0.79 | 2H | 13 | 48077 | 0.16 | 3H | 13 | 48077 | 0.16 | 4H | 13 | 48077 | 0.16 |
| 76800 | 0H | 13 | 76923 | 0.16 | 0H | 33 | 75758 | –1.36 | 0H | 65 | 76923 | 0.16 | 1H | 65 | 76923 | 0.16 |
| 115200 | 0H | 9 | 111111 | –3.55 | 1H | 11 | 113636 | –1.36 | 0H | 43 | 116279 | 0.94 | 0H | 87 | 114943 | –0.22 |
| 153600 | – | – | – | – | 1H | 8 | 156250 | 1.73 | 0H | 33 | 151515 | –1.36 | 1H | 33 | 151515 | –1.36 |
| 312500 | – | – | – | – | 0H | 8 | 312500 | 0 | 1H | 8 | 312500 | 0 | 2H | 8 | 312500 | 0 |
| <R> 625000 | – | – | – | – | 0H | 4 | 625000 | 0 | 1H | 4 | 625000 | 0 | 2H | 4 | 625000 | 0 |

备注　TPS63 ～ TPS60:　时钟选择寄存器 6 (CKSR6)的 3 ～ 0 位 (用于设置基本时钟($f_{XCLK6}$))

k:　使用波特率发生器控制寄存器 6 (BRGC6)的 MDL67 ～ MDL60 位设置的值

<R>　(k = 4, 5, 6, ..., 255)

$f_{PRS}$:　外围硬件时钟频率

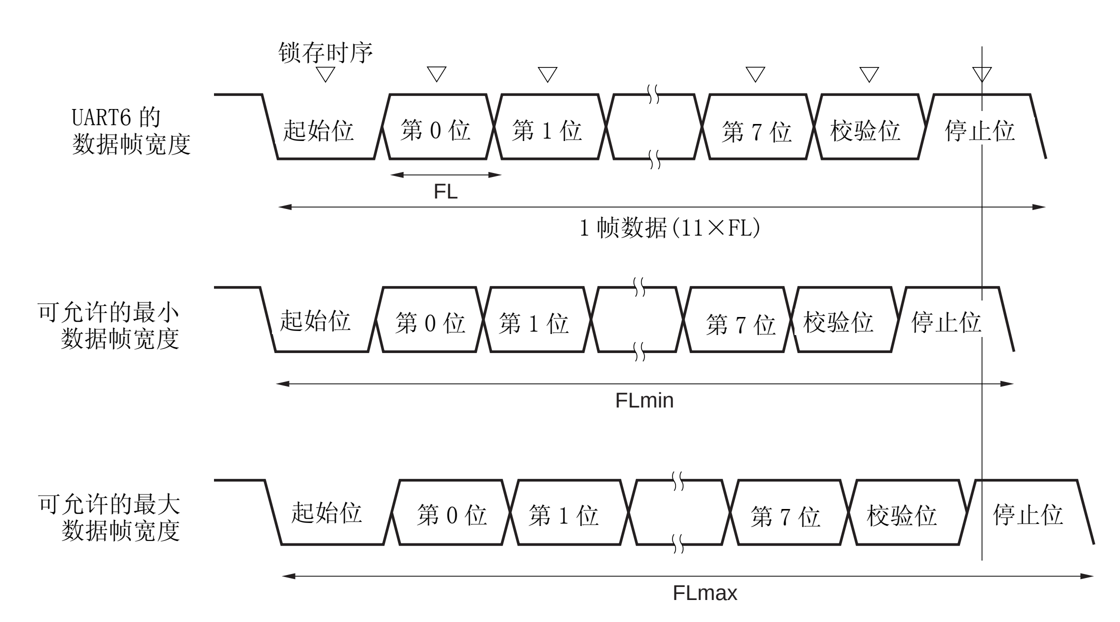
ERR:　波特率误差

**(4)** 接收期间允许的波特率范围

接收期间来自发送端的波特率可允许的误差如下所示。

**注意事项　必须确保接收期间波特率误差在允许的误差范围内，可用以下公式计算。**

**图 14-25. 接收期间允许的波特率范围**




如图 14-25 所示，当检测到起始位后，接收数据的锁存时序由波特率发生器控制寄存器 6(BRGC6)设置的计数器确定。如果数据的最后一位(停止位)满足该锁存时序，则数据可被正确接收。

假定接收数据为 11 位，各项理论值计算如下。

$$FL = (Brate)^{-1}$$

Brate:　UART6 的波特率

k:　BRGC6 的设置值

FL:　1 位数据宽度

锁存时序的极限: 2 个时钟

可允许的最小数据帧宽度: $FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k}\ FL$

因此在发送端设定的可接收的最大波特率如下所示。

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k+2}\ Brate$$

同样，可计算允许的最大数据帧宽度。

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k}\ FL$$

$$FLmax = \frac{21k-2}{20k}\ FL \times 11$$

因此在发送端设定的可接收的最小波特率如下所示。

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k-2}\ Brate$$

可通过上述最小和最大波特率公式计算在 UART6 与发送端之间允许的波特率误差，如下所示。

**表 14-6. 允许的最大/最小波特率误差**

<R>

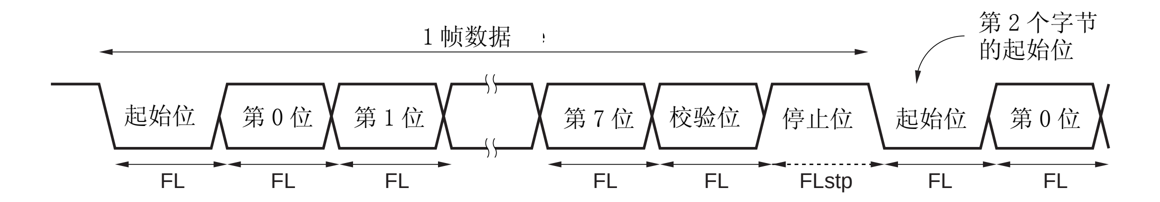
| 分频比(k) | 允许的最大波特率误差 | 允许的最小波特率误差 |
|---|---|---|
| 4 | +2.33% | –2.44% |
| 8 | +3.53% | –3.61% |
| 20 | +4.26% | –4.31% |
| 50 | +4.56% | –4.58% |
| 100 | +4.66% | –4.67% |
| 255 | +4.72% | –4.73% |

**备注**
1. 接收时允许的误差与每帧的位数、输入时钟频率和分频比(k)有关。输入时钟频率和分频比(k)越高，允许的误差就越大。
2. k: BRGC6 设置的值。

**(5)　连续发送期间数据帧的宽度**

当连续发送数据时，从停止位到下一个起始位的数据帧宽度比正常值延长了两个基本时钟。不过，由于在检测到起始位时接收端的时序被初始化，因此通信结果不会受到影响。

**图 14-26. 连续发送期间数据帧的宽度**




当 1 位数据宽度为 FL、停止位宽度为 FLstp，基本时钟频率为 $f_{XCLK6}$ 时，有以下公式成立。

$$FLstp = FL + 2/f_{XCLK6}$$

因此，连续发送期间的数据帧宽度为：

$$数据帧宽度 = 11 \times FL + 2/f_{XCLK6}$$

## 15.1 串行接口 CSI10 的功能

串行接口 CSI10 有以下两种模式。

- 操作停止模式
- 3 线串行 I/O 模式

**(1) 操作停止模式**

在不进行串行通信时可采用此模式，以降低功率消耗。

需要了解详细信息，可参见 **15.4.1 操作停止模式。**

**(2) 3 线串行 I/O 模式（可选择 MSB-first / LSB-first）**

该模式采用三条线（一条串行时钟线（$\overline{\text{SCK10}}$）和两条串行数据线（SI10 和 SO10））传输 8 位数据。

在 3 线串行 I/O 模式中数据通信时间可以缩短，因为发送和接收操作可以同步进行。

此外，可以指定采用 MSB-first 还是 LSB-first 传送 8 位数据，因此该接口可以连接任何设备。

3 线串行 I/O 模式用来连接采用时钟串行接口的外部 IC 和显示控制器。

需要了解详细信息，可参见 **15.4.2 3 线串行 I/O 模式。**

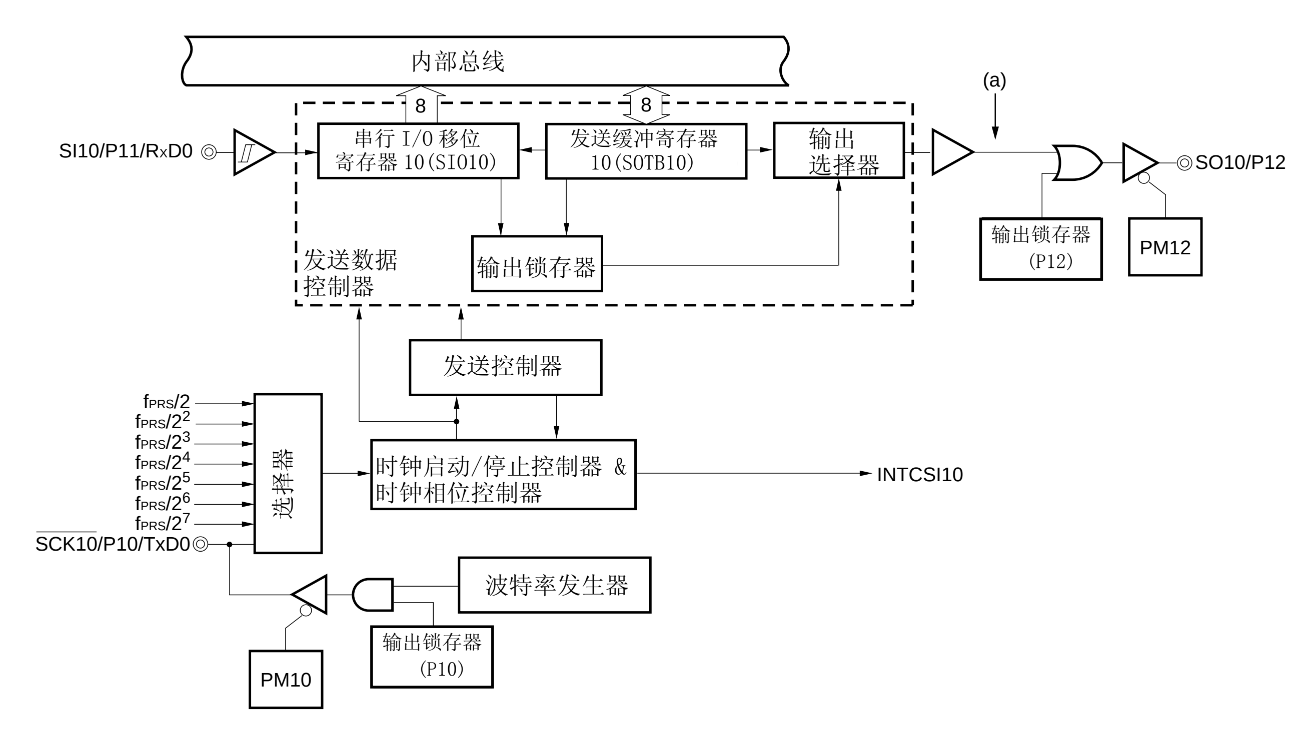
## 15.2 串行接口 CSI10 的配置

串行接口 CSI10 包括以下硬件。

表 15-1. 串行接口 CSI10 的配置

| 项目 | 配置 |
| --- | --- |
| 控制器 | 发送控制器<br>时钟启动/停止控制器 & 时钟相位控制器 |
| 寄存器 | 发送缓冲寄存器 10 （SOTB10）<br>串行 I/O 移位寄存器 10 （SIO10） |
| 控制寄存器 | 串行操作模式寄存器 10 （CSIM10）<br>串行时钟选择寄存器 10 （CSIC10）<br>或端口模式寄存器 1 （PM1）<br>端口寄存器 1 （P1） |

图 15-1. 串行接口 CSI10 的框图




备注 (a): SO10 输出

**(1) 发送缓冲寄存器 10 (SOTB10)**

该寄存器用于设置发送数据。

当串行操作模式寄存器 10（CSIM10）的第 7 位（CSIE10）和第 6 位（TRMD10）均为 1 时，通过将数据写入 SOTB10 来启动发送/接收操作。

使用串行 I/O 移位寄存器 10 将写入 SOTB10 的并行数据转换成串行数据，并把数据输出到串行输出引脚（SO10）。

可由 8 位存储器操作指令对 SOTB10 进行读写。

复位信号的产生将该寄存器清零(00H)。

**注意事项　当 CSOT10 = 1（在串行通信期间）时不要访问 SOTB1n。**

**(2) 串行 I/O 移位寄存器 10 (SIO10)**

该 8 位寄存器可将并行数据转换成串行数据，反之亦然。

可由 8 位存储器操作指令对寄存器进行读操作。

如果串行操作模式寄存器 10 （CSIM10）的第 6 位（TRMD10）= 0，则从 SIO10 读取数据可启动接收操作。

接收期间，将数据从串行输入引脚（SI10）读到 SIO10 中。

复位信号的产生将该寄存器清零（00H）。

**注意事项　当 CSOT10 = 1（在串行通信期间）时不要访问 SIO10。**

## 15.3 控制串行接口 CSI10 的寄存器

串行接口 CSI10 由以下四个寄存器控制。

- 串行操作模式寄存器 10（CSIM10）
- 串行时钟选择寄存器 10（CSIC10）
- 端口模式寄存器 1（PM1）
- 端口寄存器 1（P1）

### (1) 串行操作模式寄存器 10 （CSIM10）

CSIM10 用于选择操作模式、允许或禁止操作。

可由 1 位或 8 位存储器操作指令设置 CSIM10。

复位信号的产生将该寄存器清零（00H）。

**图 15-3. 串行操作模式寄存器 10（CSIM10）的格式**

地址: FF80H　复位后: 00H　R/W[注 1]

| 符号 | <7> | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CSIM10 | CSIE10 | TRMD10 | 0 | DIR10 | 0 | 0 | 0 | CSOT10 |

| CSIE10 | 3 线串行 I/O 模式中的操作控制 |
|---|---|
| 0 | 禁止操作[注 2] 且异步复位内部电路[注 3] |
| 1 | 允许操作 |

| TRMD10[注 4] | 发送/接收模式控制 |
|---|---|
| 0[注 5] | 接收模式（禁止发送） |
| 1 | 发送/接收模式 |

| DIR10[注 6] | 规定起始位 |
|---|---|
| 0 | MSB |
| 1 | LSB |

| CSOT10 | 通信状态标志 |
|---|---|
| 0 | 通信停止 |
| 1 | 通信正在进行中 |

注
1. 第 0 位只读。
2. 若将 P10/$\overline{\text{SCK10}}$/TxD0 和 P12/SO10 用作通用端口，设置 CSIC10 为默认状态(00H)。
3. 对 CSIM10 的第 0 位（CSOT10）和串行 I/O 移位寄存器 10 （SIO10）进行复位。
4. 当 CSOT10 = 1（在串行通信期间）时不要重写 TRMD10。
5. 当 TRMD10 = 0 时，SO10 输出(参见图 **15-1(a)**)恒为低电平。当从 SIO10 读取数据时，启动接收操作。
6. 当 CSOT10 = 1（在串行通信期间）时不要重写 DIR10。

**注意事项　第 5 位必须清零。**

**(2) 串行时钟选择寄存器 10 (CSIC10)**

该寄存器规定数据发送/接收的时序，并设置串行时钟。

可由 1 位或 8 位存储器操作指令设置 CSIC10。

复位信号的产生将寄存器清零（00H）。

**图 15-5. 串行时钟选择寄存器 10（CSIC10）的格式**

地址: FF81H　复位后: 00H　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CSIC10 | 0 | 0 | 0 | CKP10 | DAP10 | CKS102 | CKS101 | CKS100 |

| CKP10 | DAP10 | 规定数据发送/接收的时序 | 类型 |
|---|---|---|---|
| 0 | 0 | $\overline{SCK10}$<br>SO10 D7 D6 D5 D4 D3 D2 D1 D0<br>SI10 输入时序 | 1 |
| 0 | 1 | $\overline{SCK10}$<br>SO10 D7 D6 D5 D4 D3 D2 D1 D0<br>SI10 输入时序 | 2 |
| 1 | 0 | $\overline{SCK10}$<br>SO10 D7 D6 D5 D4 D3 D2 D1 D0<br>SI10 输入时序 | 3 |
| 1 | 1 | $\overline{SCK10}$<br>SO10 D7 D6 D5 D4 D3 D2 D1 D0<br>SI10 输入时序 | 4 |

| CKS102 | CKS101 | CKS100 | CSI10 串行时钟的选择 | | | | | 模式 |
|---|---|---|---|---|---|---|---|---|
| | | | | $f_{PRS}$ = 2 MHz | $f_{PRS}$ = 5 MHz | $f_{PRS}$ = 10 MHz | $f_{PRS}$ = 20 MHz | |
| 0 | 0 | 0 | $f_{PRS}/2$ | 1 MHz | 2.5 MHz | 5 MHz | 10 MHz | 主设备模式 |
| 0 | 0 | 1 | $f_{PRS}/2^2$ | 500 kHz | 1.25 MHz | 2.5 MHz | 5 MHz | |
| 0 | 1 | 0 | $f_{PRS}/2^3$ | 250 kHz | 625 kHz | 1.25 MHz | 2.5 MHz | |
| 0 | 1 | 1 | $f_{PRS}/2^4$ | 125 kHz | 312.5 kHz | 625 kHz | 1.25 MHz | |
| 1 | 0 | 0 | $f_{PRS}/2^5$ | 62.5 kHz | 156.25 kHz | 312.5 kHz | 625 kHz | |
| 1 | 0 | 1 | $f_{PRS}/2^6$ | 31.25 kHz | 78.13 kHz | 156.25 kHz | 312.5 kHz | |
| 1 | 1 | 0 | $f_{PRS}/2^7$ | 15.63 kHz | 39.06 kHz | 78.13 kHz | 156.25 kHz | |
| 1 | 1 | 1 | 输入到 $\overline{SCK10}$ 的外部时钟 | | | | | 从设备模式 |

注意事项 **1.** 当 **CSIE10=1** 时（允许操作），不要对 **CSIC10** 进行写操作。

**2.** 若将 **P10/$\overline{SCK10}$/TxD0** 和 **P12/SO10** 用作通用端口，设置 **CSIC10** 为默认状态**(00H)**。

**3.** 复位后数据时钟的相位类型为类型 **1**。

备注　$f_{PRS}$: 外围硬件时钟频率

**(3) 端口模式寄存器 1 （PM1）**

这两个寄存器用于按位设置端口 0 和 1 的输入/输出操作模式。

当 P10/SCK10/TxD0 和作为串行接口的时钟输出引脚时，对 PM10 清零，并将 P10 的输出锁存器的值设置为 1。

当 P12/SO10 作为串行接口的数据输出引脚时，则对 PM12 清零，并将 P12 的输出锁存器的值设置为 0。

当 P10/SCK10/TxD0 作为串行接口的时钟输入引脚、P11/SI10/RxD0 作为数据输入引脚时，将 PM10、PM11 设置为 1。此时，P10 和 P11 的输出锁存值可能是 0 或 1。

可由 1 位或 8 位存储器操作指令设置 PM1。

复位信号的产生将这两个寄存器的值设置为 FFH。

**图 15-4. 端口模式寄存器 1（PM1）的格式**

地址：FF21H　　　复位后：FFH　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM1 | PM17 | PM16 | PM15 | PM14 | PM13 | PM12 | PM11 | PM10 |

| PM1n | P1n 引脚 I/O 模式选择（n=0～7） |
|---|---|
| 0 | 输出模式（输出缓冲器打开） |
| 1 | 输入模式（输出缓冲器关闭） |

## 15.4 串行接口 CSI10 的操作

串行接口 CSI10 可在以下两种模式中使用。

- 操作停止模式
- 3 线串行 I/O 模式

### 15.4.1 操作停止模式

在此模式下，不能进行串行通信，因此，可降低功率消耗。此外，P10/$\overline{\text{SCK10}}$/TxD0，P11/SI10/RxD0，P12/SO10 在此模式下可作为通用 I/O 端口引脚使用。

**(1) 使用的寄存器**

由串行操作模式寄存器 10（CSIM10）设置操作停止模式。

通过将 CSIM10 的第 7 位（CSIE10）清零，设置该模式。

**(a) 串行操作模式寄存器 10 (CSIM10)**

可由 1 位或 8 位存储器操作指令设置 CSIM10。

复位信号的产生将 CSIM10 清零（00H）。

地址: FF80H　复位后: 00H　R/W

| 符号 | <7> | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CSIM10 | CSIE10 | TRMD10 | 0 | DIR10 | 0 | 0 | 0 | CSOT10 |

| CSIE10 | 3 线串行 I/O 模式中的操作控制 |
|---|---|
| 0 | 禁止操作[注 1]，并异步复位内部电路[注 2] |

注　**1.**　若将 P10/$\overline{\text{SCK10}}$/TxD0 和 P12/SO10 用作通用端口，设置 CSIC10 为默认状态(00H)。

　　**2.**　对 CSIM10 的第 0 位 （CSOT10）和串行 I/O 移位寄存器 10 （SIO10）进行复位。

### 15.4.2 3 线串行 I/O 模式

3 线串行 I/O 模式用来连接外部 IC 和具有时钟串行接口的显示控制器。

该模式中使用三条线进行通信: 串行时钟（$\overline{\text{SCK10}}$）、串行输出（SO10）和串行输入（SI10）线。

**(1) 使用的寄存器**

- 串行操作模式寄存器 10（CSIM10）
- 串行时钟选择寄存器 10（CSIC10）
- 端口模式寄存器 1（PM1）
- 端口寄存器 1（P1）

3 线串行 I/O 模式设置操作的基本步骤如下。

<1>　设置 CSIC10 寄存器（参见 **图 15-3**）
<2>　设置 CSIM10 寄存器的 0 第 4 和 6 位(DIR10 和 TRMD10)（参见 **图 15-2**）。
<3>　CSIM10 的第 7 位（CSIE1n）置 1。 → 允许发送/接收。
<4>　将数据写入发送缓冲寄存器 10（SOTB10）。 → 启动发送/接收操作。
　　　从串行 I/O 移位寄存器 10（SIO10）中读取数据。 → 启动数据接收操作。

**注意事项　　在设置端口模式寄存器和端口寄存器时，要考虑与通信另一方的关系。**

寄存器设置与引脚之间的关系如下所示。

**表 15-2.** 寄存器设置与引脚之间的关系

| CSIE10 | TRMD10 | PM11 | P11 | PM12 | P12 | PM10 | P10 | CSI10 操作 | 引脚功能 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | SI10/RxD0/P11 | SO10/P12 | $\overline{SCK10}$/TxD0/P10 |
| 0 | × | ×[注 1] | ×[注 1] | ×[注 1] | ×[注 1] | ×[注 1] | ×[注 1] | 停止 | RxD0/P11 | P12 | TxD0/P10[注 2] |
| 1 | 0 | 1 | × | ×[注 1] | ×[注 1] | 1 | × | 从设备接收[注 3] | SI10 | P12 | $\overline{SCK10}$ (输入)[注 3] |
| 1 | 1 | ×[注 1] | ×[注 1] | 0 | 0 | 1 | × | 从设备发送[注 3] | RxD0/P11 | SO10 | $\overline{SCK10}$ (输入)[注 3] |
| 1 | 1 | 1 | × | 0 | 0 | 1 | × | 从设备发送/接收[注 3] | SI10 | SO10 | $\overline{SCK10}$ (输入)[Note 3] |
| 1 | 0 | 1 | × | ×[注 1] | ×[注 1] | 0 | 1 | 主设备接收 | SI10 | P12 | $\overline{SCK10}$ (输出) |
| 1 | 1 | ×[注 1] | ×[注 1] | 0 | 0 | 0 | 1 | 主设备发送 | RxD0/P11 | SO10 | $\overline{SCK10}$ (输出) |
| 1 | 1 | 1 | × | 0 | 0 | 0 | 1 | 主设备发送/接收 | SI10 | SO10 | $\overline{SCK10}$ (输出) |

注　**1.**　可用作端口引脚。
　　**2.**　如果 P10/$\overline{SCK10}$/TxD0 用作端口引脚，则将 CKP10 清零。
　　**3.**　如果使用从设备模式，则须将 CKS102、CKS101 和 CKS100 设置为 1、1 和 1。

**备注**

| | |
|---|---|
| ×: | 不必考虑 |
| CSIE10: | 串行操作模式寄存器 10 （CSIM10）的第 7 位 |
| TRMD10: | CSIM10 的第 6 位 |
| CKP10: | 串行时钟选择寄存器 10 （CSIC10）的第 4 位 |
| CKS102， CKS101， CKS100: | CSIC10 的第 0 ～ 2 位 |
| PM1×: | 端口模式寄存器 |
| P1×: | 端口输出锁存器 |

**(2) 通信操作**

在 3 线串行 I/O 模式中，是以 8 位为单元发送或接收数据。每位数据与串行时钟同步地发送或接收。

当串行操作模式寄存器 10（CSIM10）的第 6 位（TRMD10）=1 时，允许发送或接收数据。当数据被写入到发送缓冲寄存器 10（SOTB10）时，开始进行发送/接收。此外，串行操作模式寄存器 10 (CSIM10)的第 6 位(TRMD10) = 0 时，允许接收数据。

当数据从串行 I/O 移位寄存器 10（SIO10）被读出时，启动接收操作。

通信开始后，CSIM10 的第 0 位（CSOT1n）=1。当 8 位数据通信结束时，设置通信完成中断请求标志（CSIIF10），且对 CSOT10 清零。这样允许进行下一个通信。

**注意事项　　当 CSOT10 = 1 时（串行通信期间），不要访问控制寄存器和数据寄存器。**

**图 15-5. 3 线串行 I/O 模式中的时序 (1/2)**

**(a) 发送/接收时序 (类型 1: TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 0)**

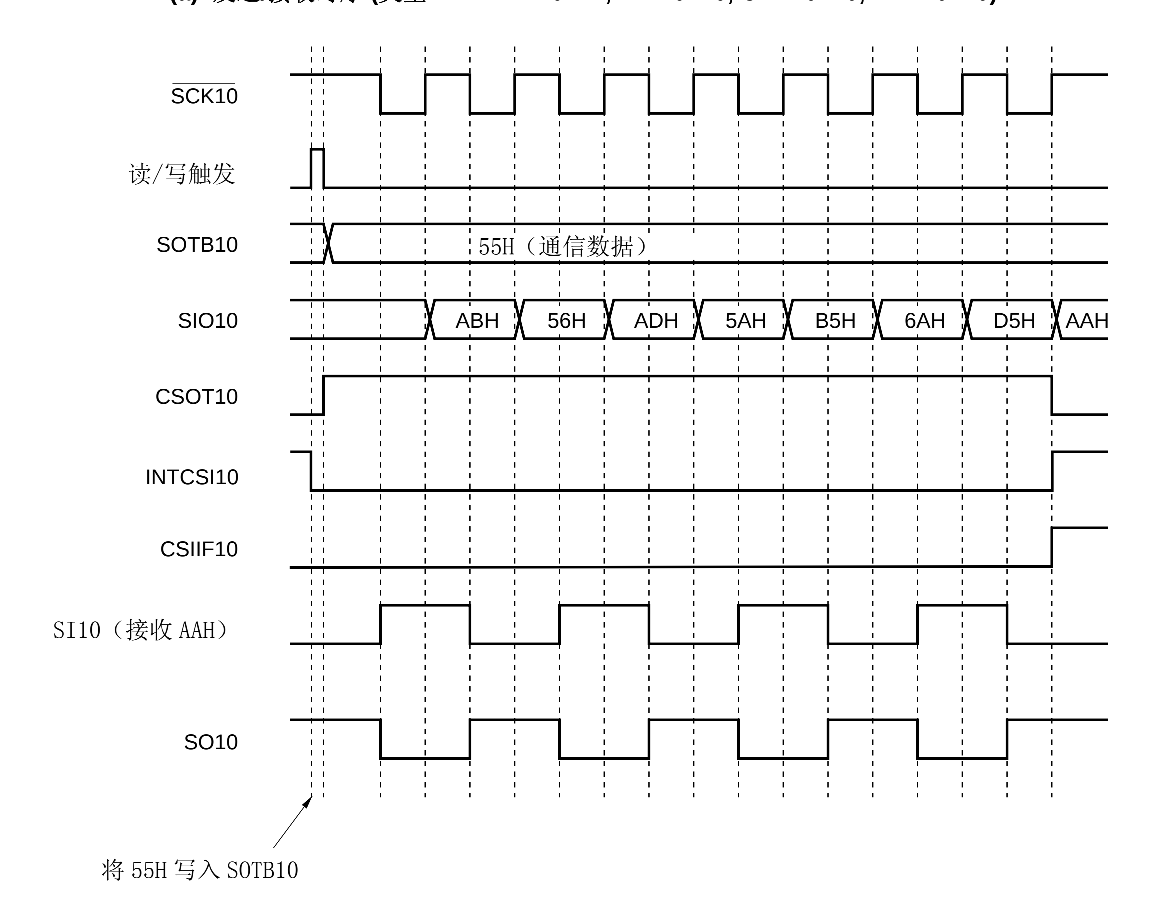


**图 15-5. 3 线串行 I/O 模式中的时序 (2/2)**

**(b) 发送/接收时序 (类型 2: TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 1)**

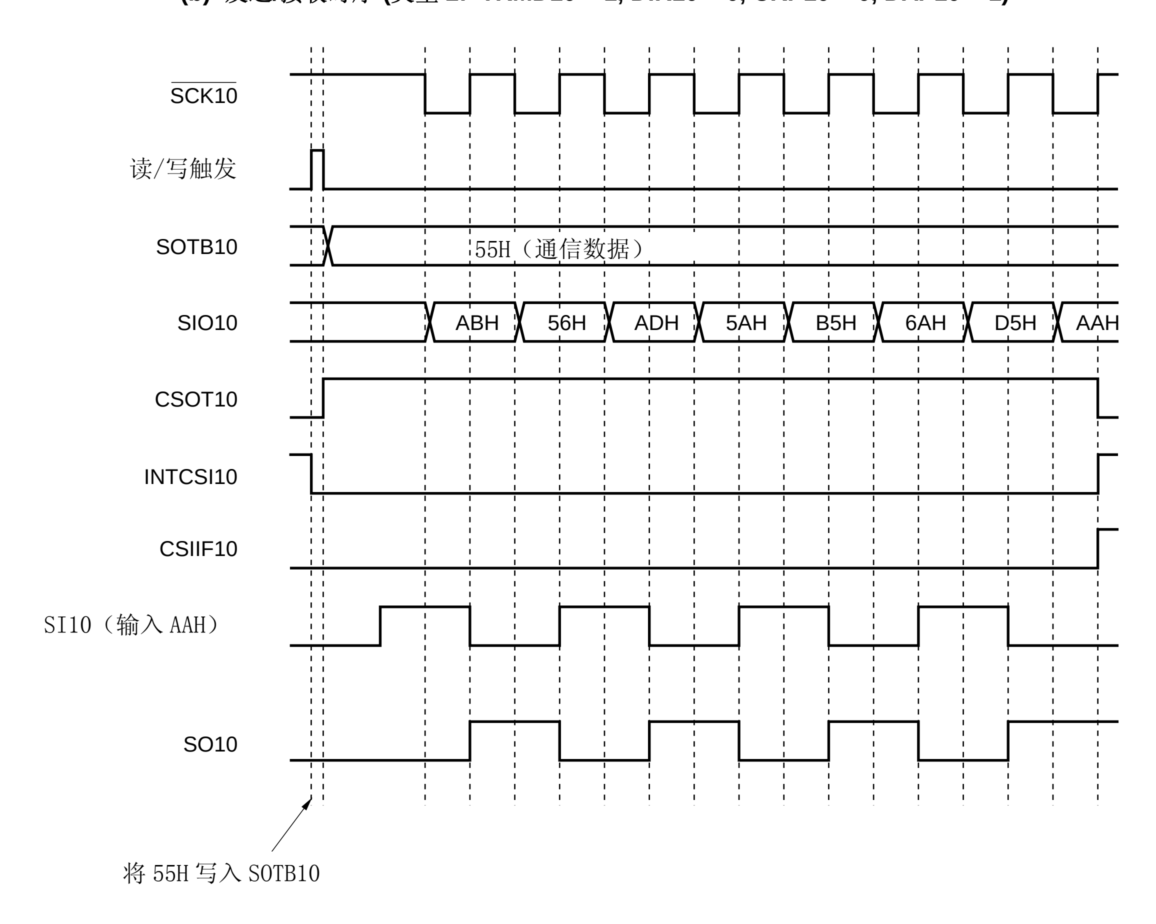
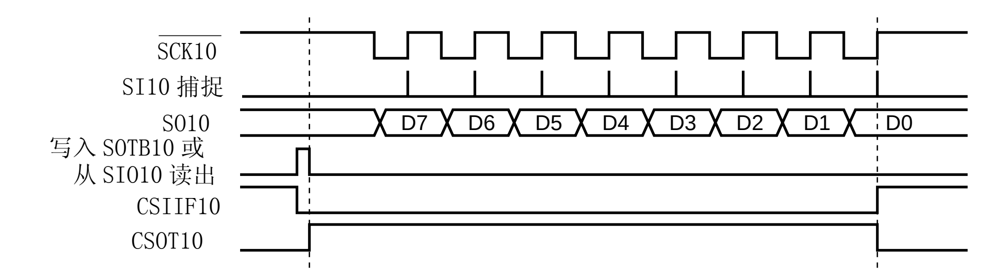
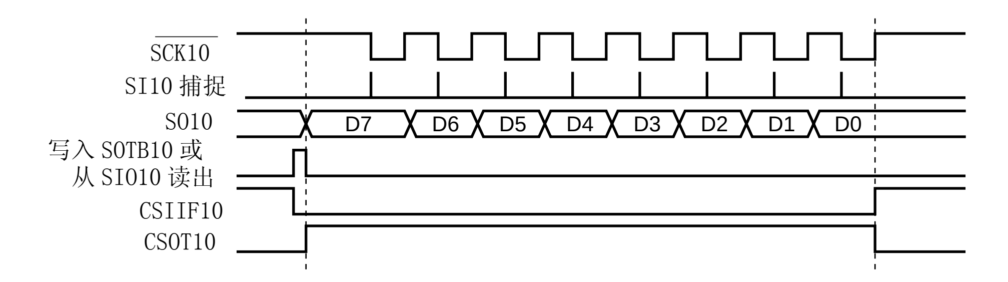


图 **15-6.** 时钟/数据相位的时序

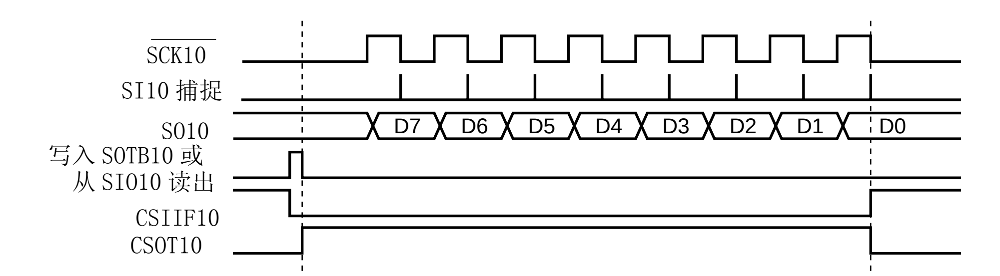
**(a)** 类型 **1: CKP10 = 0, DAP10 = 0, DIR10 = 0**



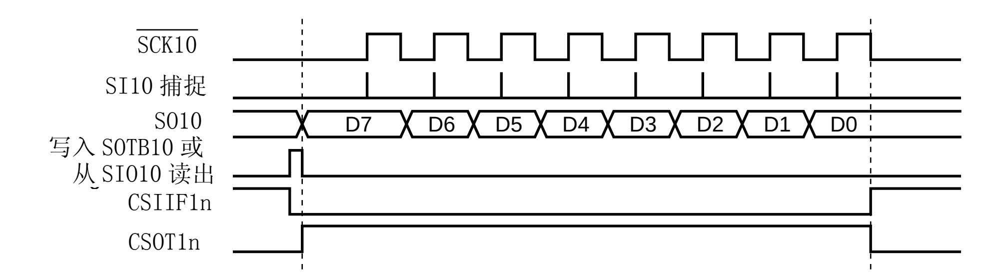

**(b)** 类型 **2: CKP10 = 0, DAP10 = 1, DIR10 = 0**




**(c)** 类型 **3: CKP10 = 1, DAP10 = 0, DIR10 = 0**




**(d)** 类型 **4: CKP10 = 1, DAP10 = 1, DIR10 = 0**




**备注**　　上图显示了一个通信操作过程，即采用 MSB first 发送数据。

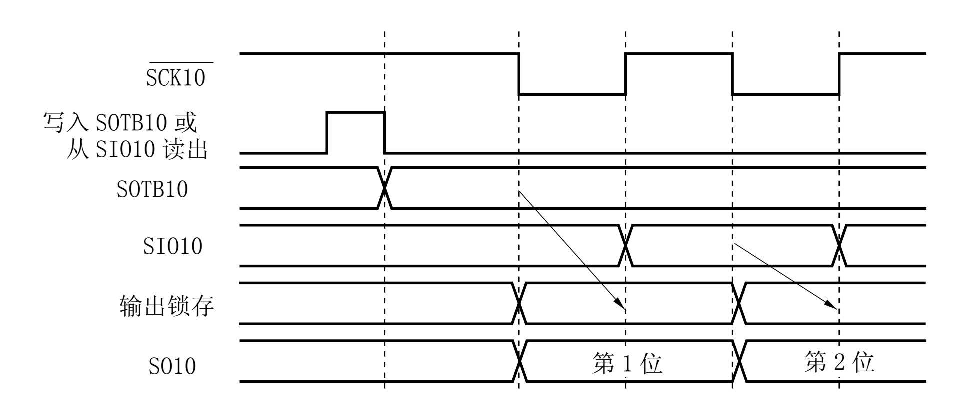
**(3) 输出到 SO10 引脚的时序 （起始位）**

通信开始后，将发送缓冲寄存器 10（SOTB10）的值从 SO10 引脚输出。

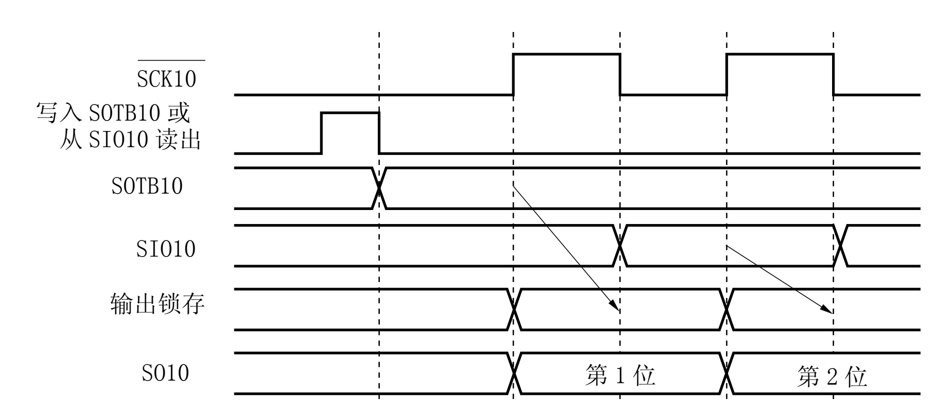
此时起始位的输出操作如下所示。

**图 15-7. 起始位的输出操作 (1/2)**

**(a) 类型 1: CKP10 = 0, DAP10 = 0**




**(b) 类型 3: CKP10 = 1, DAP10 = 0**

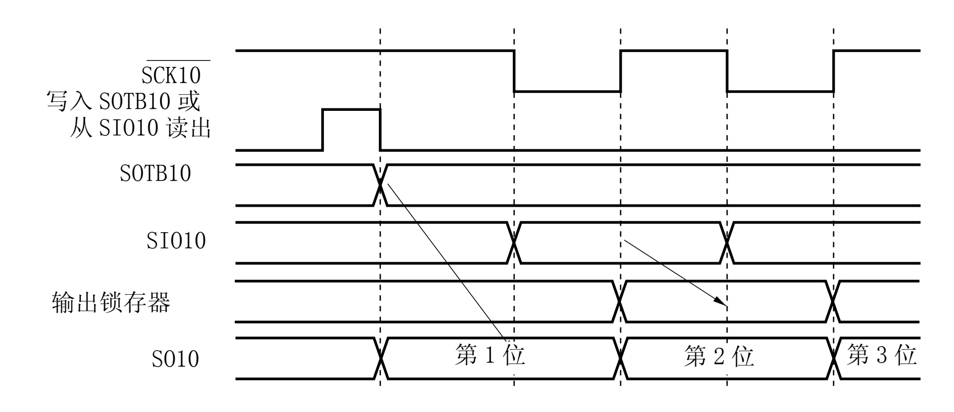



在 $\overline{\text{SCK10}}$ 的下降沿（或上升沿），起始位数据通过 SOTB10 寄存器被直接锁存到输出锁存器，并通过输出选择器从 SO10 引脚输出。然后在 $\overline{\text{SCK10}}$ 下一个上升沿（或下降沿）将 SOTB10 寄存器的内容传送到 SIO10 中，并移出 1 位。同时，通过 SI10 引脚将接收数据的起始位存储到 SIO10 寄存器中。

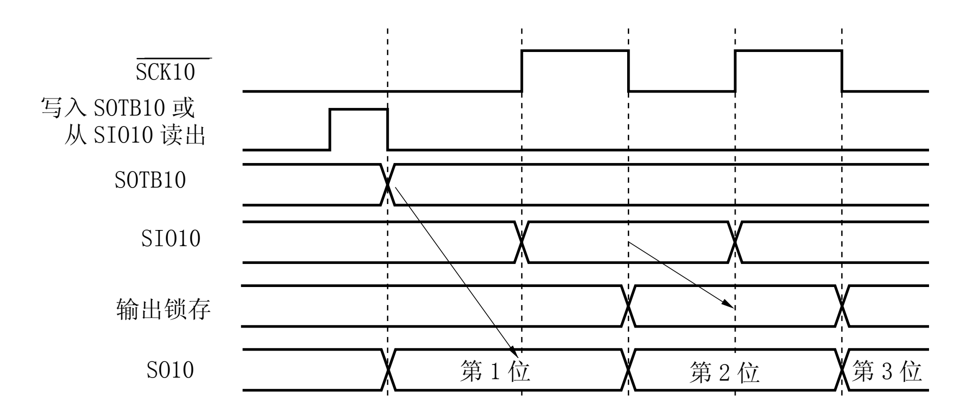
在下一个 $\overline{\text{SCK10}}$ 的下降沿（或上升沿），通过 SIO10 寄存器将第 2 位以及随后各位锁存到输出锁存器，然后从 SO10 引脚输出数据。

图 15-7. 起始位的输出操作 (2/2)

(c) 类型 2: CKP10 = 0, DAP10 = 1




(d) 类型 4: CKP10 = 1, DAP10 = 1




在 SOTB10 的写信号或 SIO10 的读信号的下降沿，起始位数据通过 SOTB10 寄存器被直接锁存，并通过输出选择器从 SO10 引脚输出。然后在 SCK10 下一个下降沿（或上升沿）将 SOTB10 寄存器的内容传送到 SIO10 中，并移出 1 位。同时，将接收数据的起始位通过 SI10 引脚存储到 SIO10 寄存器。

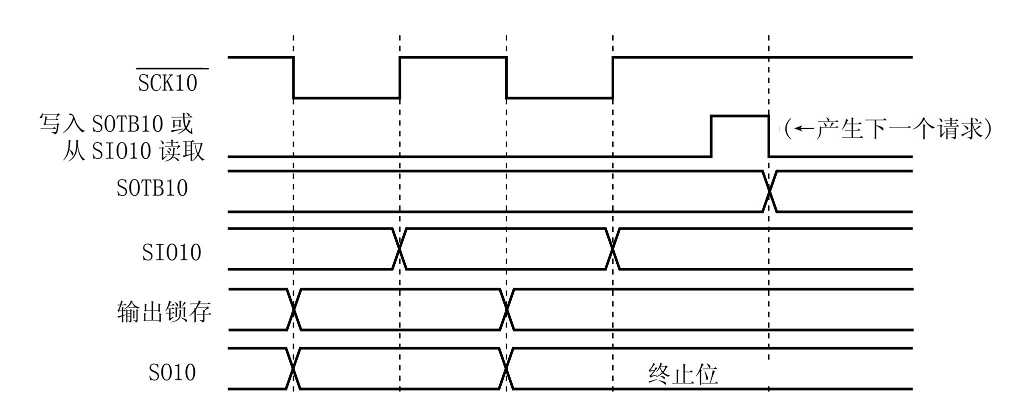
在下一个 SCK10 的上升沿（或下降沿），通过 SIO10 寄存器将第 2 位以及随后各位锁存到输出锁存器，然后从 SO10 引脚输出数据。

**(4)　SO10** 引脚的输出值 （终止位）

在通信完成后，**SO10** 引脚保存终止位的输出值。

图 **15-8. SO10** 引脚的输出值 **(**终止位**) (1/2)**

**(a)** 类型 **1: CKP10 = 0, DAP10 = 0**




**(b)** 类型 **3: CKP10 = 1, DAP10 = 0**

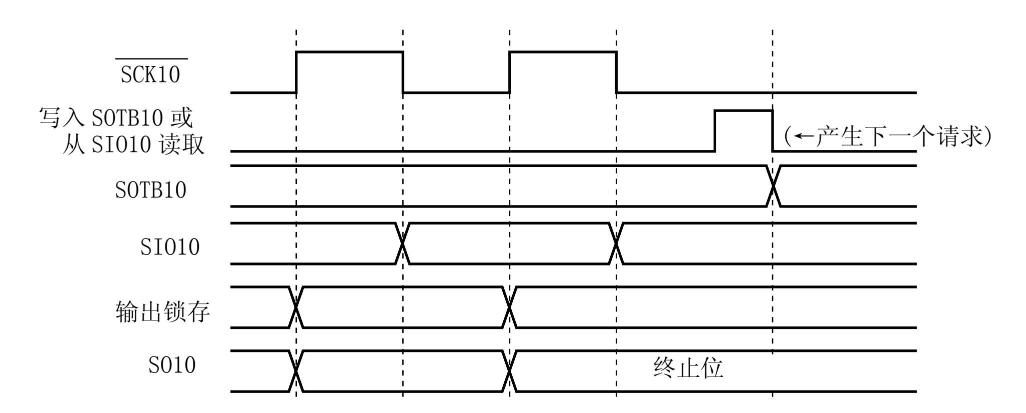
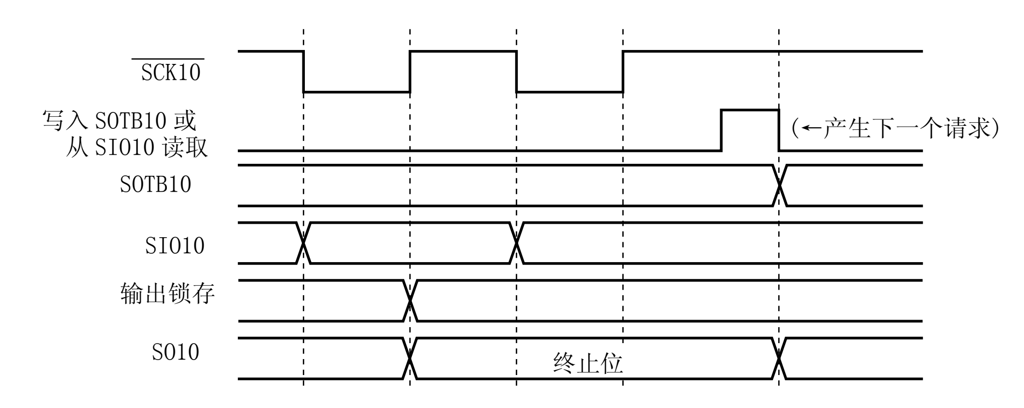
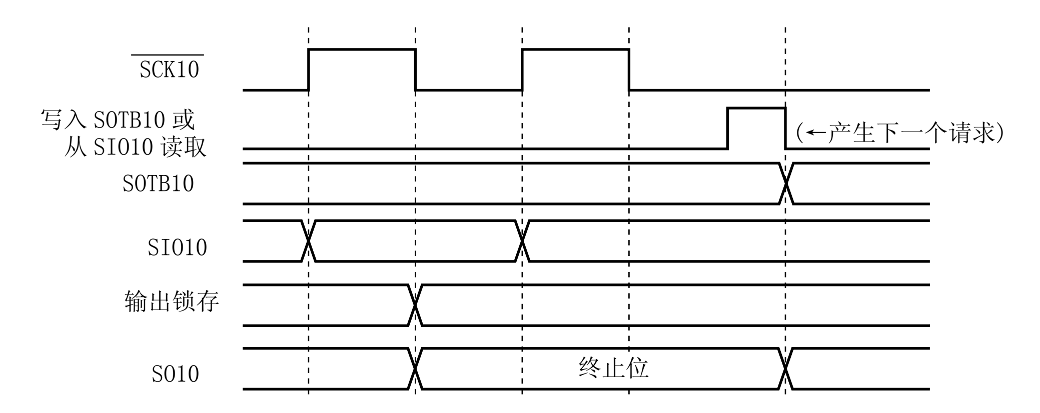


**图 15-8. SO10 引脚的输出值 (终止位) (2/2)**

**(c) 类型 2: CKP10 = 0, DAP10 = 1**




**(d) 类型 4: CKP10 = 1, DAP10 = 1**



**(5) SO10 输出 (参看 图 15-1 中的 (a))**

如果串行操作模式寄存器 10（CSIM10）的第 7 位（CSIE1n）被清零，则 SO10 输出状态如下。

**表 15-3. SO10 输出状态**

| TRMD10 | DAP10 | DIR10 | SO10 输出[注 1] |
|---|---|---|---|
| TRMD10 = 0[注 2] | – | – | 输出低电平[注 2] |
| TRMD10 = 1 | DAP10 = 0 | – | SO10 锁存值<br>(低电平输出) |
| | DAP10 = 1 | DIR10 = 0 | SOTB10 第 7 位的值 |
| | | DIR10 = 1 | SOTB10 第 0 位的值 |

注　**1.** 根据 PM12 和 P12 以及 SO10 的输出，决定 SO10/P12 引脚的实际输出。
**2.** 复位后的状态

注意事项　如果对 **TRMD10**、**DAP10** 和 **DIR10** 进行写操作，则 **SO10** 的输出值将发生改变。

# 第十六章　串行接口 IIC0

## 16.1 串行接口 IIC0 的功能

串行接口 IIC0 有以下两种模式。

**(1) 操作停止模式**

在不进行串行传送时可采用此模式，以降低功耗。

**(2) $I^2C$ 总线模式 (支持多个主设备)**

该模式采用两条线(一条串行时钟(SCL0)总线和一条串行数据总线(SDA0))进行多设备的 8 位数据传送。

该模式遵循$I^2C$总线格式并且主设备能够通过串行数据总线，产生“起始条件”，“地址”，“传送方向指示”，“数据”和“停止条件”等信号传送至从设备。从设备能够通过硬件自动地检测这些接收到的状态和数据。这个功能可以简化控制$I^2C$总线的应用程序部分。

由于 SCL0 和 SDA0 引脚是开漏输出，IIC0 总线的串行时钟线和串行数据总线需要连接上拉电阻。

图 16-1 是串行接口 IIC0 的框图。

**图 16-1.  串行接口 IIC0 框图**

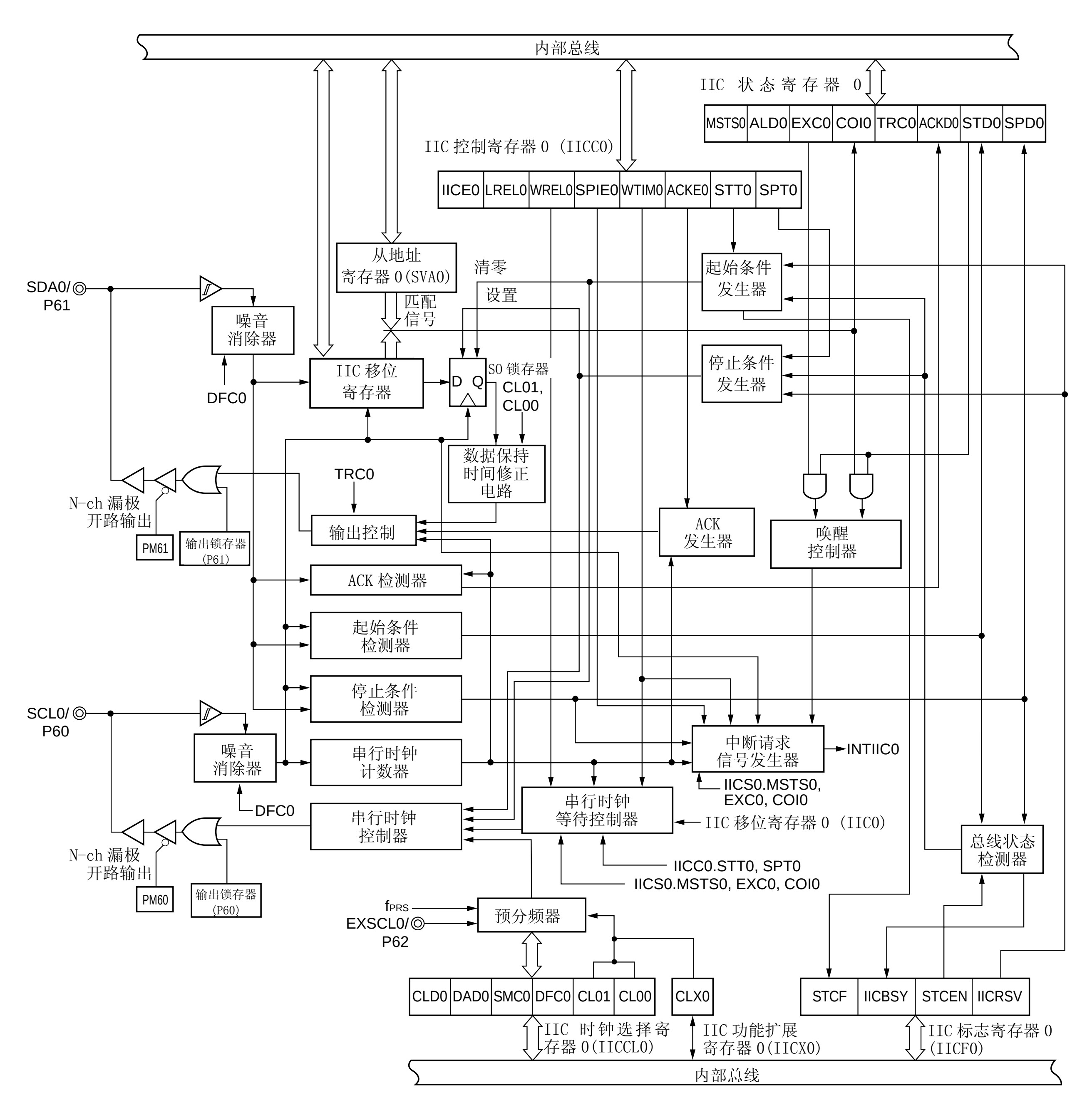
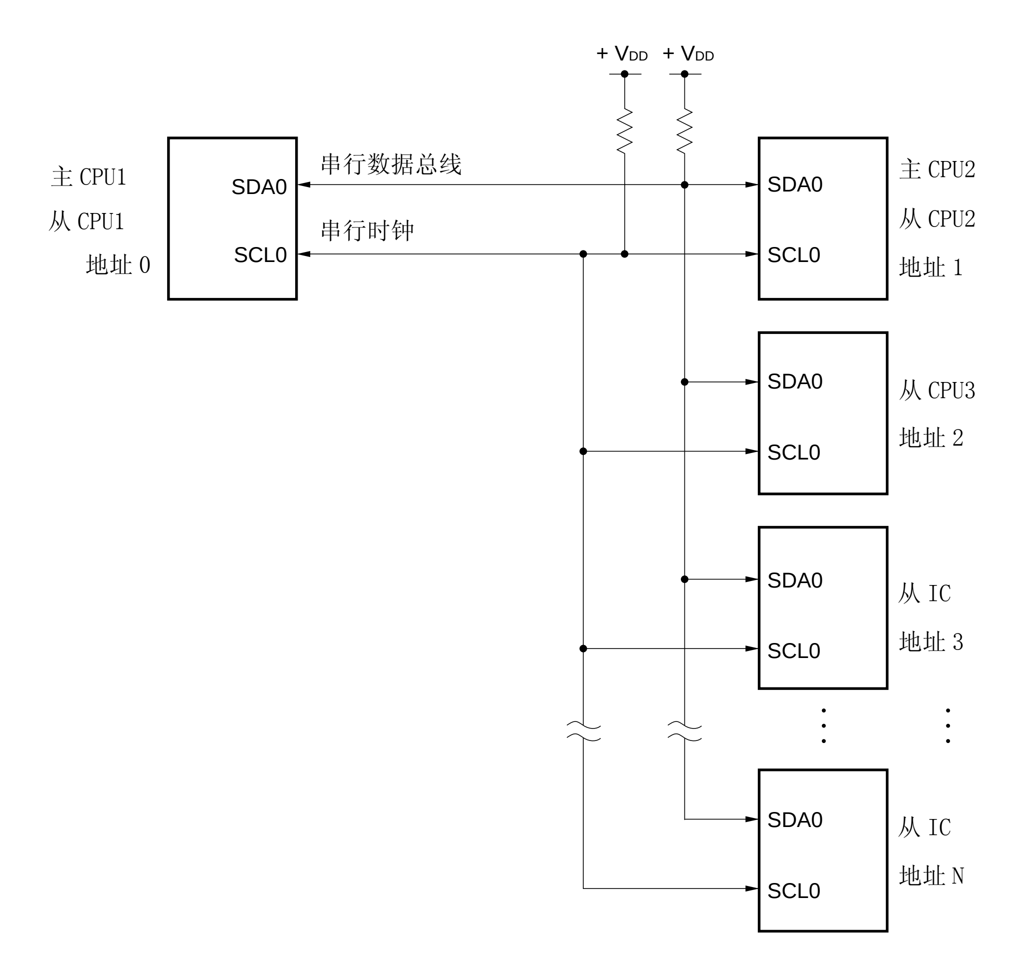


图 16-2 是串行总线配置示例。

**图 16-2. 使用$I^2C$ 总线的串行总线配置示例**



## 16.2 串行接口 IIC0 的配置

串行接口 IIC0 包括下列硬件。

**表 16-1. 串行接口 IIC0 配置**

| 项目 | 配置 |
|---|---|
| 寄存器 | IIC 移位寄存器 0(IIC0)<br>从设备地址寄存器 0(SVA0) |
| 控制寄存器 | IIC 控制寄存器 0(IICC0)<br>IIC 状态寄存器 0(IICS0)<br>IIC 标志寄存器 0(IICF0)<br>IIC 时钟选择寄存器 0(IICCL0)<br>IIC 功能扩展寄存器 0(IICX0)<br>端口模式寄存器 6(PM6)<br>端口寄存器 6(P6) |

**(1) IIC 移位寄存器 0(IIC0)**

IIC0 用于 8 位串行数据和 8 位并行数据的相互转换，并且和串行时钟同步。IIC0 可以用来发送和接收数据。

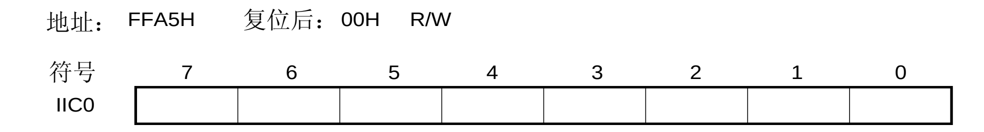
对 IIC0 的读写操作可以控制实际的发送和接收操作。

在等待期间通过对 IIC0 写入数据，可以取消等待状态并起始数据传送操作。

可由 8 位存储器操作指令设置 IIC0。

复位信号将 IIC0 清零。

**图 16-3. IIC 转换寄存器 0 的格式 (IIC0)**




注意事项 **1.** **在数据传送期间不要对 IIC0 进行写操作。**

**2.** **仅在等待期间才能对 IIC0 进行读写操作。除等待期间外，禁止在其他的通信状态下访问 IIC0。然而，作为主设备时，在通信触发位(STT0)被设为 1 之后 IIC0 只能被写入一次。**

**(2) 从设备地址寄存器 0(SVA0)**

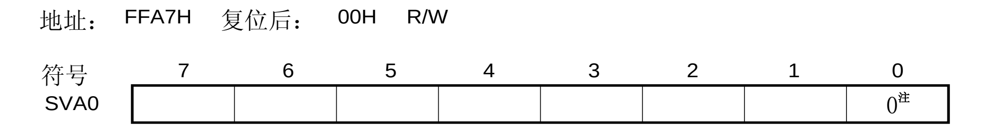
该寄存器用于在从设备模式下存储本地地址。

可由 8 位存储器操作指令设置 SVA0。

但是当 STD0=1 时，禁止重写该寄存器(当检测到起始条件时)。

复位信号将 SVA0 清零。

**图 16-4. 从地址寄存器 0 的格式 (SVA0)**




注　　第 0 位恒为 0。

**(3) SO 锁存器**

SO 锁存器用来保持 SDA0 引脚的输出电平。

**(4) 唤醒控制器**

当该寄存器接收的地址与从设备地址寄存器 0(SVA0)设置的地址相等，或者当接收到一个扩展代码时，这个电路将产生一个中断请求信号(INTIIC0)。

**(5) 预分频器**

用于选择采样时钟。

**(6) 串行时钟计数器**

该计数器对发送/接收操作期间输出或输入的串行时钟进行计数，同时用于验证 8 位数据是否已发送或接收。

**(7) 中断请求信号发生器**

此电路用于控制中断请求信号(INTIIC0)的产生。

一个 $I^2C$ 中断请求由下列两个触发条件产生：

- 串行时钟的第 8 或第 9 个下降沿(通过 WTIM0 位设置)
- 当检测到停止条件(通过 SPIE0 位设置)时

备注　WTIM0 位：IIC 控制寄存器 0 (IICC0)的第 3 位
　　　SPIE0 位： IIC 控制寄存器 0 (IICC0)的第 4 位

**(8) 串行时钟控制器**

在主设备模式下，此电路通过 SCL0 引脚，从采样时钟产生时钟输出。

**(9) 串行时钟等待控制器**

用于控制等待时序。

**(10) ACK 信号发生器，停止条件检测器，起始条件检测器和 ACK 信号检测器**

这些电路用于产生和检测每种状态。

**(11) 数据保持时间修正电路**

此电路根据串行时钟的下降沿产生数据保持时间。

**(12) 起始条件发生器**

当 STT0 位为 1 时，该电路产生一个启动条件信号。

但是，在禁止通信预约的状态下(IICRSV 位 = 1)，只要总线尚未被释放(IICBSY 位 = 1)，起始条件请求将被忽略并且 STCF 被置为 1。

**(13) 停止条件发生器**

当 SPT0 位为 1 时，此电路产生一个停止条件。

**(14) 总线状态检测器**

此电路通过检测起始条件和停止条件来确定总线是否被释放。

然而，如果操作后总线状态无法被立即检测到，则初始状态由 STCEN 位来设置。

备注　STT0 位：　IIC 控制寄存器 0 (IICC0)的第 1 位
SPT0 位：　IIC 控制寄存器 0 (IICC0)的第 0 位
IICRSV 位：IIC 标志寄存器 0 的第 0 位
IICBSY 位：IIC 标志寄存器 0 的第 6 位
STCF 位：　IIC 标志寄存器 0 的第 7 位
STCEN 位：IIC 标志寄存器 0 的第 1 位

## 16.3 控制串行接口 IIC0 的寄存器

串行接口 IIC0 由下列 7 个寄存器控制：

- IIC 控制寄存器 0(IICC0)
- IIC 标志寄存器 0 (IICF0)
- IIC 状态寄存器 0 (IICS0)
- IIC 时钟选择寄存器 0 (IICCL0)
- IIC 功能扩展寄存器 0 (IICX0)
- 端口模式寄存器 6 (PM6)
- 端口寄存器 6 (P6)

**(1) IIC 控制寄存器 0 (IICC0)**

这个寄存器用于允许/停止$I^2C$ 的操作、设置等待时间和设置$I^2C$ 的其他操作。

可由 1 位或 8 位存储器操作指令对 IICC0 进行设置。但必须在 IICE0 = 0 或者在等待期间设置 SPIE0，WTIM0，和 ACKE0 位。当 IICE0 位从“0”变为“1”时，这些位可以同时被设置。

复位信号将 IICC0 清零。

**图 16-5. IIC 控制寄存器 0 (IICC0)的格式 (1/4)**

地址： FFA6H　　复位后： 00H　　R/W

| 符号 | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| IICC0 | IICE0 | LREL0 | WREL0 | SPIE0 | WTIM0 | ACKE0 | STT0 | SPT0 |

| IICE0 | I$^2$C 操作允许 |
|---|---|
| 0 | 停止操作。复位IIC 状态寄存器 0(IICS0)[注1]。停止内部操作。 |
| 1 | 允许操作 |
| 当 SCL0 和 SDA0 在高电平时，必须将该位设置为 1。 | |
| 清零条件 (IICE0 = 0) | 设置条件 (IICE0 = 1) |
| • 通过指令清零<br>• 复位 | • 通过指令设置 |

| LREL0[注2] | 退出通信 |
|---|---|
| 0 | 正常操作 |
| 1 | 此时从当前的通信模式退出，进入待机模式。当执行此操作后设置的值自动清零。<br>用于那些已接受了本地无关扩展码的情况。<br>SCL0 和 SDA0 线被设置为高阻抗状态。<br>下列 IIC 控制器 0 (IICC0) 和 IIC 状态寄存器 0 (IICS0)的标志位被清零。<br>• STT0　• SPT0　• MSTS0　• EXC0　• COI0　• TRC0　• ACKD0　• STD0 |
| 从通信模式退出后一直保持待机模式，直到满足下列通信条件。<br>• 检测到停止条件后，在主设备模式下重新启动。<br>• 启动后，发生地址相等或接收到扩展码。 | |
| 清零条件 (LREL0 = 0) | 设置条件 (LREL0 = 1) |
| • 执行后自动清零<br>• 复位 | • 通过指令设置 |

| WREL0[注2] | 等待取消 |
|---|---|
| 0 | 不取消等待 |
| 1 | 取消等待。在等待取消后，此设置被自动清除。 |
| 在第 9 个时钟脉冲的下降沿有效启动的发送状态(TRC0 = 1)下，在等待期间设置 WREL0(等待取消)时，SDA0 将进入高阻抗状态(TRC0 = 0)。 | |
| 清零条件 (WREL0 = 0) | 设置条件 (WREL0 = 1) |
| • 执行后自动清零<br>• 复位 | • 通过指令设置 |

注　**1.** 对 IICS0 寄存器、IICF0 寄存器的 STCF0 位与 IICBSY 位和 IICCL0 寄存器的 CLD0 位与 DAD0 位进行复位。

　　**2.** 当 IICE0 = 0 时，本标志信号无效。

注意事项　**允许I$^2$C 操作(IICE0 = 1)后，在SCL0 为高电平并且SDA0 为低电平时，立即检测到起始条件。允许I$^2$C操作(IICE0 = 1)后，立即通过 1 位存储器操作指令设置LREL0(1)。**

### 图 16-5. IIC 控制寄存器 0 (IICC0)的格式 (2/4)

<table>
<tr><th>SPIE0[注 1]</th><th colspan="2">当检测到停止条件时，允许/禁止产生中断请求</th></tr>
<tr><td>0</td><td colspan="2">禁止</td></tr>
<tr><td>1</td><td colspan="2">允许</td></tr>
<tr><td colspan="2">清零条件 (SPIE0 = 0)</td><td>设置条件 (SPIE0 = 1)</td></tr>
<tr><td colspan="2">• 通过指令清零<br>• 复位</td><td>• 通过指令进行设置</td></tr>
</table>

<table>
<tr><th>WTIM0[注 1]</th><th colspan="2">对等待和中断请求产生的控制</th></tr>
<tr><td>0</td><td colspan="2">在第 8 个时钟的下降沿产生中断请求。<br>主设备模式：输出 8 个时钟之后，将时钟输出设置为低电平且设置等待状态。<br>从设备模式：输入 8 个时钟之后，将时钟设置为低电平且为主设备设置等待状态。</td></tr>
<tr><td>1</td><td colspan="2">在第 9 个时钟的下降沿产生中断请求。<br>主设备模式：输出 9 个时钟之后，将时钟输出设置为低电平且设置等待状态。<br>从设备模式：输入 9 个时钟之后，将时钟设置为低电平且设置主设备等待状态。</td></tr>
<tr><td colspan="3">地址传送期间，在第 9 个时钟的下降沿产生中断，而与该位的设置无关。地址传送完成之后该位设置才有效。主设备模式下，在地址传送期间的第 9 个时钟的下降沿插入一个等待信号。对于已接收到本地地址的从设备，在发出应答信号(　　)之后的第 9 个时钟的下降沿插入一个等待信号。但是，当从设备已经收到一个扩展码时，将在第 8 个时钟的下降沿插入等待信号。</td></tr>
<tr><td colspan="2">清零条件 (WTIM0 = 0)</td><td>设置条件 (WTIM0 = 1)</td></tr>
<tr><td colspan="2">• 通过指令清零<br>• 复位</td><td>• 通过指令进行设置</td></tr>
</table>

<table>
<tr><th>ACKE0[注 1,2]</th><th colspan="2">应答控制</th></tr>
<tr><td>0</td><td colspan="2">禁止应答</td></tr>
<tr><td>1</td><td colspan="2">允许应答。在第 9 个时钟期间，将 SDA0 设置为低电平。但是，在非扩展模式及地址传送期间，$\overline{ACK}$ 信号无效。</td></tr>
<tr><td colspan="2">清零条件 (ACKE0 = 0)</td><td>设置条件 (ACKE0 = 1)</td></tr>
<tr><td colspan="2">• 通过指令进行清零<br>• 复位</td><td>• 通过指令进行设置</td></tr>
</table>

注　**1.** IICE0 = 0 时，此标志信号无效。

　　**2.** 在地址传送期间及编码为非扩展码时，所设的值无效。
　　在从设备模式下并且地址相等时，无论设置何值都将产生应答信号。

**图 16-5. IIC 控制寄存器 0 (IICC0) 的格式 (3/4)**

<table>
<tr><td>STT0[注]</td><td colspan="2">启动条件触发</td></tr>
<tr><td>0</td><td colspan="2">不产生起始条件</td></tr>
<tr><td>1</td><td colspan="2">总线释放时(STOP 模式下)：<br>产生一个起始条件(作为主设备启动)。SCL0 处于高电平时，SDA0 从高电平变为低电平时产生启动条件。然后，经过额定时间，SCL0 变为低电平。<br>有第三方通信时：<br>• 当允许通信预约时 (IICRSV = 0)<br>该位作为启动条件预约标志。设置为 1 时，在总线释放之后自动产生一个起始条件。<br>• 当禁止通信预约时 (IICRSV = 1)<br>将 STCF 设置为 1，且 STT0 被清零。不产生起始条件。<br>等待状态(主设备模式)：<br>退出等待状态之后产生一个重复起始条件。</td></tr>
<tr><td colspan="3">设置时序的注意事项<br>• 主设备接收：传送期间不能设置为 1。仅在 ACKE0 被清零且从设备通报结束接收时的等待期间，才能设为 1。<br>• 主设备发送：应答期间内，不能正常产生起始条件。在第 9 个时钟输出后的等待期间可以设置为 1。<br>• 不能和 SPT0 同时设为 1。<br>• 禁止在 STT0 被清零前重复设置为 1。</td></tr>
<tr><td colspan="2">清零条件 (STT0 = 0)</td><td>设置条件 (STT0 = 1)</td></tr>
<tr><td colspan="2">• 禁止通信预约时，通过将 SST0 设置为 1 进行清零<br>• 由仲裁失败清零<br>• 通过主设备产生起始条件后进行清零<br>• 由 LREL0＝1(退出通信)清零<br>• 当 IICE0 = 0 (操作停止)时<br>• 复位</td><td>• 通过指令进行设置</td></tr>
</table>

注　IICE0 = 0 时，该标志信号无效。

备注　**1.** 第 1 位 (STT0)在设置后再读取时将会变为 0。
**2.** IICRSV：IIC 标志寄存器 (IICF0)的第 0 位
STCF：　IIC 标志寄存器(IICF0)的第 7 位

**图 16-5. IIC 控制寄存器 0 (IICC0)的格式 (4/4)**

| SPT0 | 停止条件触发器 |
| --- | --- |
| 0 | 不产生停止条件 |
| 1 | 产生停止条件 (结束主设备传送)<br>SDA0 变为低电平后，设置 SCL0 为高电平或等待其变为高电平。然后，经过额定时间后，SDA0 从低电平变为高电平，则产生一个停止条件。 |

<table>
<tr><td colspan="2">设置时序注意事项<br>• 主设备接收：　传送期间不能设为 1。<br>仅在 ACKE0 被清零且从设备通报结束接收时的等待期间，才能设为 1。<br>• 主设备发送：　应答期间内，不能正常产生停止条件。在第 9 个时钟输出后的等待期间可以设为 1。<br>• 不能和 SST0 同时设置为 1<br>• 仅在主设备模式[注]下，SPT0 才能设置为 1<br>• WTIM0 清零后，如果在输出 8 个时钟之后的等待期间将 SPT0 设为 1，那么注意，在第 9 个时钟的高电平期间将产生一个停止条件。在输出 8 个时钟后的等待期间 WTIM0 应从 0 变为 1，并且在输出第 9 个时钟后的等待期间 SPT0 应被设置为 1。<br>•禁止在 SPT0 被清零前重复设置为 1。</td></tr>
<tr><td>清零条件 (SPT0 = 0)</td><td>设置条件 (SPT0 = 1)</td></tr>
<tr><td>• 由仲裁失败清零<br>• 检测到停止条件后自动清零<br>• 可由 LREL0＝1(退出通信)清零<br>• IICE0 = 0 时(操作停止)<br>• 复位</td><td>• 通过指令进行设置</td></tr>
</table>

注　仅在主设备模式下 SPT0 可设置为 1。但是，在切换到允许操作状态后检测到第 1 个停止条件之前必须将 SPT0 设置为 1，产生停止条件。如需了解详细信息，可参见 **16.5.15 其他注意事项**。

**注意事项　在第 9 个时钟，当 IIC 状态寄存器 0(IICS0)的第 3 位(TRC0)被设为 1 和 WREL0 设置为 1 时，，取消等待状态后，TRC0 被清零，SDA0 线被设为高阻抗状态。**

备注　第 0 位 (SPT0)在设置后再读取时将变为 0。

**(2) IIC 状态寄存器 0 (IICS0)**

此寄存器用于表示I²C的状态。

仅当 STT0 = 1 且在等待期间，才能通过 1 位或 8 位存储器操作指令读取 IICS0。

复位信号的产生将 IICS0 设置为 00H。

**注意事项** 如果从 **IICS0** 中读取数据，将产生一个等待周期。当 **CPU** 使用副系统时钟且外围硬件时钟停止时，不要从 **IICS0** 中读取数据。详见 第三十二章 等待注意事项 。

**图 16-6. IIC 状态寄存器 0 (IICS0) 的格式 (1/3)**

地址： FFAAH 复位后： 00H R

| 符号 | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| IICS0 | MSTS0 | ALD0 | EXC0 | COI0 | TRC0 | ACKD0 | STD0 | SPD0 |

| MSTS0 | 主设备状态 |
|---|---|
| 0 | 从设备状态或通信待机状态 |
| 1 | 主设备通信状态 |

| 清零条件 (MSTS0 = 0) | 设置条件 (MSTS0 = 1) |
|---|---|
| • 检测到停止条件时<br>• ALD0＝1 时(仲裁失败)<br>• 由 LREL0 = 1 (退出通信)清零<br>• IICE0 从 1 变为 0 (操作停止)时<br>• 复位 | • 起始条件产生时 |

| ALD0 | 仲裁失败的检测 |
|---|---|
| 0 | 此状态表示没有仲裁或仲裁结果为"获胜"。 |
| 1 | 此状态表示仲裁结果为"失败"。MSTS0 被清零。 |

| 清零条件 (ALD0 = 0) | 设置条件 (ALD0 = 1) |
|---|---|
| •读取[注] IICS0 之后自动清零<br>• IICE0 从 1 变为 0 (操作停止)时<br>• 复位 | • 仲裁结果为"失败"时。 |

| EXC0 | 扩展码接收的检测 |
|---|---|
| 0 | 未接收到扩展码 |
| 1 | 接收到扩展码 |

| 清零条件 (EXC0 = 0) | 设置条件 (EXC0 = 1) |
|---|---|
| • 检测到起始条件时<br>• 检测到停止条件时<br>• 由 LREL0 = 1 (退出通信)清零<br>• IICE0 从 1 变为 0 (操作停止)时<br>• 复位 | • 当接收到的地址数据的高 4 位为"0000"或"1111"时(在第 8 个时钟的上升沿设置) |

注 即使使用 1 位存储器操作指令没有对 IICS0 进行操作，此寄存器也会清零。因此当使用 ALD0 位时，应先读取该位数据。

备注 LREL0： IIC 控制寄存器 0(IICC0)的第 6 位
IICE0：IIC 控制寄存器 0(IICC0)的第 7 位

### 图 16-6. IIC 状态寄存器 0 (IICS0) 的格式 (2/3)

| COI0 | 地址相等的检测 |
|---|---|
| 0 | 地址不相等 |
| 1 | 地址相等 |
| 清零条件 (COI0 = 0) | 设置条件 (COI0 = 1) |
| • 检测到起始条件时<br>• 检测到停止条件时<br>• 由 LREL0 = 1 (退出通信)清零<br>• IICE0 从 1 变为 0 (操作停止)时<br>• 复位 | • 当接收地址和本地地址(从设备地址寄存器 0 (SVA0)的内容)相等时<br>(在第 8 个时钟的上升沿设置) |

| TRC0 | 发送/接收状态的检测 |
|---|---|
| 0 | 接收状态(非发送状态) 。SDA0 线被设为高阻抗状态。 |
| 1 | 发送状态。允许将 SO0 锁存器中的内容输出到 SDA0 线 (在第 1 个字节的第 9 个时钟的下降沿才有效)。 |
| 清零条件 (TRC0 = 0) | 设置条件 (TRC0 = 1) |
| <主设备模式和从设备模式><br>• 检测到停止条件时<br>• 由 LREL0 = 1(退出通信)清零<br>• IICE0 从 1 变为 0 (操作停止)时<br>• 由WREL0 = 1[注] (取消等待)清零<br>• ALD0 从 0 变为 1 (仲裁失败)时<br>• 复位<br><主设备模式><br>• 当“1”输出到第 1 个字节的 LSB (传送方向指示位)时<br><从设备模式><br>• 检测到启动条件时<br>• 当 “0” 输入到第 1 个字节的 LSB (传送方向指示位)时<br><未用于通信时> | <主设备模式><br>• 产生起始条件时<br>• 当“0”输出到第 1 个字节的 LSB (传送方向指示位)时<br><从设备模式><br>• 当“1”输入到第 1 个字节的 LSB (传送方向指示位)时 |

**注**　当 IIC 状态寄存器 0(IICS0)的第 3 位(TRC0)为 1 时，通过在第 9 个时钟将 IIC 控制寄存器 0(IICC0)的第 5 位(WREL0)设置为 1，来取消等待状态，TRC0 被清零且 SDA0 线进入高阻抗状态。

**备注**　LREL0： IIC 控制寄存器 0 (IICC0)的第 6 位
IICE0： IIC 控制寄存器 0 (IICC0)的第 7 位

**图 16-6. IIC 状态寄存器 0 (IICS0)的格式(3/3)**

<table>
<tr><td>ACKD0</td><td colspan="2">应答信号的检测 ($\overline{ACK}$)</td></tr>
<tr><td>0</td><td colspan="2">未检测到应答信号</td></tr>
<tr><td>1</td><td colspan="2">已检测到应答信号</td></tr>
<tr><td colspan="2">清零条件 (ACKD0 = 0)</td><td>设置条件 (ACKD0 = 1)</td></tr>
<tr><td colspan="2">• 检测到停止条件时<br>• 在下一个字节的第一个时钟的上升沿<br>• 由 LREL0 = 1 (退出通信)清零<br>• 当 IICE0 从 1 变为 0 (操作停止)时<br>• 复位</td><td>• 在 SCL0 的第 9 个时钟的上升沿将 SDA0 线设为低电平之后</td></tr>
</table>

<table>
<tr><td>STD0</td><td colspan="2">启动条件的检测</td></tr>
<tr><td>0</td><td colspan="2">未检测到起始条件</td></tr>
<tr><td>1</td><td colspan="2">已检测到起始条件。这表示地址传送阶段有效。</td></tr>
<tr><td colspan="2">清零条件 (STD0 = 0)</td><td>设置条件 (STD0 = 1)</td></tr>
<tr><td colspan="2">• 检测到停止条件时<br>• 在地址发送之后的下一个字节的第 1 个时钟的上升沿<br>• 由 LREL0 = 1 (退出通信)清零<br>• 当 IICE0 从 1 变为 0 (操作停止)时<br>• 复位</td><td>• 检测到起始条件时</td></tr>
</table>

<table>
<tr><td>SPD0</td><td colspan="2">停止条件的检测</td></tr>
<tr><td>0</td><td colspan="2">未检测到停止条件</td></tr>
<tr><td>1</td><td colspan="2">已检测到停止条件。主设备通信结束，释放总线。</td></tr>
<tr><td colspan="2">清零条件 (SPD0 = 0)</td><td>设置条件 (SPD0 = 1)</td></tr>
<tr><td colspan="2">• 在设置该位并检测到起始条件后的地址发送字节第 1 个时钟的上升沿<br>• 当 IICE0 从 1 变为 0 (操作停止)时<br>• 复位</td><td>• 检测到停止条件时</td></tr>
</table>

**备注** LREL0： IIC 控制寄存器 0(IICC0)的第 6 位

IICE0： IIC 控制寄存器 0(IICC0)的第 7 位

**(3) IIC 标志寄存器 0 (IICF0)**

此寄存器用于设置$I^2C$的操作模式，并指示$I^2C$ 总线的状态。

IICF0 可由 1 位或 8 位的存储器操作指令进行设置。但是，STCF 和 IICBSY 位是只读的。

IICRSV 位用于允许/禁止通信预约功能 (详见 **16.5.14 通信预约**)。

STCEN 用于设置 IICBSY 的初始值 (详见 **16.5.15 其他注意事项**)。

仅当$I^2C$操作被禁止(IIC控制寄存器 0 (IICC0)的第 7 位(IICE0) = 0)时才可对IICRSV 和STCEN位进行写操作。当允许操作时，可以读取IICF0 寄存器。

复位信号的产生将 IICF0 变为 00H。

**图 16-7. IIC 标志寄存器 0 (IICF0)的格式**

地址：FFABH　　复位后：00H　　R/W注

| 符号 | <7> | <6> | 5 | 4 | 3 | 2 | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| IICF0 | STCF | IICBSY | 0 | 0 | 0 | 0 | STCEN | IICRSV |

| STCF | STT0 清零标志 |
|---|---|
| 0 | 产生起始启动条件 |
| 1 | 未能产生起始条件：STT0 清零 |

| 清零条件（STCF=0） | 设置条件（STCF=1） |
|---|---|
| • 通过 STT0=1 清零<br>• IICE0=0 时（操作停止）<br>• 复位 | 当禁止通信预约时(IICRSV=1)，未能产生起始条件且 STT0 被清零 |

| IICBSY | $I^2C$总线状态标志 |
|---|---|
| 0 | 总线释放状态（当 STCEN0=1 时的通信初始状态） |
| 1 | 总线通信状态（当 STCEN0=0 时的通信初始状态） |

| 清零条件（IICBSY=0） | 设置条件（IICBSY=1） |
|---|---|
| • 检测到停止条件<br>• IICE0=0 时（操作停止）<br>• 复位 | • 检测到起始条件<br>• 当 STCEN=0 时设置 IICE0 |

| STCEN | 允许初始启动触发 |
|---|---|
| 0 | 允许操作后（IICE0=1），可以在检测到停止条件的基础上产生一个起始条件。 |
| 1 | 允许操作后（IICE0=1），可以在不检测到停止条件的基础上产生一个起始条件。 |

<R>

| 清零条件（STCEN=0） | 设置条件（STCEN=1） |
|---|---|
| • 检测到停止条件<br>• 复位 | • 通过指令进行设置 |

| IICRSV | 通信预约功能禁止位 |
|---|---|
| 0 | 允许通信预约 |
| 1 | 禁止通信预约 |

| 清零条件（IICRSV=0） | 设置条件（IICRSV=1） |
|---|---|
| • 通过指令进行清零<br>• 复位 | • 通过指令进行设置 |

注　第 6 位和第 7 位是只读的。

注意事项　**1.** 仅当操作停止**(IICE0 = 0)**时才可对 **STCEN** 进行写操作。

**2.** 当 **STCEN = 1** 时不论总线处于何种状态，总线释放状态**(IICBSY = 0)**都可以识别，因此当第一个起始条件产生**(STT0 = 1)**时，必须确保没有第三方通信，以防通信被破坏。

**3.** 仅当操作停止**(IICE0 = 0)**时才可对 **IICRSV** 进行写操作。

备注　STT0：IIC 控制寄存器 0 (IICC0)的第 1 位
IICE0：IIC 控制寄存器 0 (IICC0)的第 7 位

**(4) IIC 时钟选择寄存器 0 (IICCL0)**

此寄存器用来为I$^2$C总线设置传送时钟。

可由 1 位或 8 位的存储器操作指令设置IICCL0。但是，CLD0 和DAD0 位是只读的。应根据IIC功能扩展寄存器 0 (IICX0)的第 0 位(CLX0)设置SMC0、CL01 和CL00 位(详见 **16.3 (6) I$^2$C 传送时钟设置方法**)。

IIC 控制寄存器 0 (IICC0)的第 7 位(IICE0)为 0 时设置 IICCL0。

复位信号的产生将 IICCL0 设置为 00H。

**图 16-8. IIC 时钟选择寄存器 0 (IICCL0)的格式**

地址： FFA8H　　复位后： 00H R/W[注]

| 符号 | 7 | 6 | <5> | <4> | <3> | <2> | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| IICCL0 | 0 | 0 | CLD0 | DAD0 | SMC0 | DFC0 | CL01 | CL00 |

| CLD0 | SCL0 引脚电平的检测(仅当 IICE0 = 1 时有效) |
|---|---|
| 0 | SCL0 引脚在低电平被检测。 |
| 1 | SCL0 引脚在高电平被检测。 |
| 清零条件 (CLD0 = 0) | 设置条件 (CLD0 = 1) |
| • SCL0 引脚为低电平时<br>• IICE0 = 0 (操作条件)时<br>• 复位 | • SCL0 引脚为高电平时 |

| DAD0 | SDA0 引脚电平的检测(仅当 IICE0 = 1 时有效) |
|---|---|
| 0 | SDA0 引脚在低电平被检测 |
| 1 | SDA0 引脚在高电平被检测 |
| 清零条件 (DAD0 = 0) | 设置条件 (DAD0 = 1) |
| • SDA0 引脚为低电平时<br>• IICE0 = 0 (操作条件)时<br>• 复位 | • SDA0 引脚为高电平时 |

| SMC0 | 操作模式切换 |
|---|---|
| 0 | 标准模式 |
| 1 | 高速模式 |

| DFC0 | 数字滤波器操作控制 |
|---|---|
| 0 | 关闭数字滤波器 |
| 1 | 打开数字滤波器 |
| 数字滤波器只能用于高速模式<br>在高速模式下，不论 DFC0 位如何设置(设置(1)/清除(0))，传送时钟不会发生变化。<br>数字滤波器用于高速模式下的噪声消除 | |

**注** 第 4 位和第 5 位是只读的。

**备注**　IICE0： IIC 控制寄存器 0 (IICC0)的第 7 位

**(5) IIC 功能扩展寄存器 0 (IICX0)**

此寄存器用于设置$I^2C$的功能扩展。

可由 1 位或 8 位存储器操作指令设置IICX0。根据IIC时钟选择寄存器 0(IICCL0)的第 3、1 和 0 位(SMC0、CL01 和CL00)设置CLX0 (详见 **16.3 (6) $I^2C$ 传送时钟设置方法**)。

IIC 控制寄存器 0(IICC0)的第 7 位(IICE0)为 0 时设置 IICX0。

复位信号将 IICX0 清零。

**图 16-9. IIC 功能扩展寄存器 0(IICX0)的格式**

地址：FFA9H　　复位后：00H　　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| IICX0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | CLX0 |

**(6) $I^2C$ 传送时钟设置方法**

$I^2C$传送时钟的频率 ($f_{SCL}$)通过下列表达式计算。

$$f_{SCL} = 1/(m \times T + t_R + t_F)$$

m = 12, 16, 24, 44, 66, 86 (详见表 **16-2 选择时钟设置**)

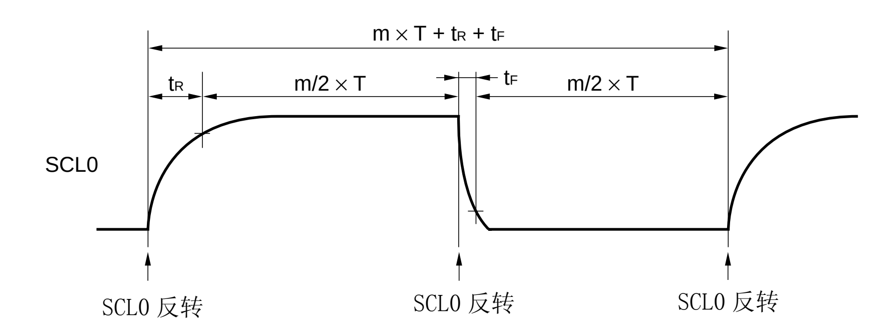
T:　$1/f_W$

$t_R$:　SCL0 上升时间

$t_F$:　SCL0 下降时间

举例来说，当 $f_W = f_{PRS}/2$ = 4.19 MHz、m = 86、$t_R$ = 200 ns和$t_F$ = 50 ns时，$I^2C$ 传送时钟频率($f_{SCL}$)通过下列表达式计算。

$f_{SCL}$ = 1/(88 × 238.7 ns + 200 ns + 50 ns) ≅ 48.1 kHz




通过 IIC 时钟选择寄存器 0(IICCL0)的第 3、1 和 0 位 (SMC0、CL01 和和 CL00)与 IIC 功能扩展寄存器 0(IICX0)的第 0 位(CLX0)设置选择时钟。

表 **16-2.** 选择时钟设置

| IICX0 | IICCL0 | | | 选择时钟 ($f_W$) | 传送时钟 ($f_W$/m) | 可设置的选择时钟($f_W$)范围 | 操作模式 |
|---|---|---|---|---|---|---|---|
| 第 0 位 | 第 3 位 | 第 1 位 | 第 0 位 | | | | |
| CLX0 | SMC0 | CL01 | CL00 | | | | |
| 0 | 0 | 0 | 0 | $f_{PRS}/2$ | $f_W/44$ | 2.00 ～ 4.19 MHz | 正常模式 (SMC0 位= 0) |
| 0 | 0 | 0 | 1 | $f_{PRS}/2$ | $f_W/86$ | 4.19 ～ 8.38 MHz | |
| 0 | 0 | 1 | 0 | $f_{PRS}/4$ | $f_W/86$ | | |
| 0 | 0 | 1 | 1 | $f_{EXSCL0}$ | $f_W/66$ | 6.4 MHz | |
| 0 | 1 | 0 | × | $f_{PRS}/2$ | $f_W/24$ | 4.00 ～ 8.38 MHz | 高速模式 (SMC0 位= 1) |
| 0 | 1 | 1 | 0 | $f_{PRS}/4$ | $f_W/24$ | | |
| 0 | 1 | 1 | 1 | $f_{EXSCL0}$ | $f_W/18$ | 6.4 MHz | |
| 1 | 0 | × | × | 禁止设置 | | | |
| 1 | 1 | 0 | × | $f_{PRS}/2$ | $f_W/12$ | 4.00 ～ 4.19 MHz | 高速模式 (SMC0 位= 1) |
| 1 | 1 | 1 | 0 | $f_{PRS}/4$ | $f_W/12$ | | |
| 1 | 1 | 1 | 1 | 禁止设置 | | | |

注意事项　**允许操作(通过设置IIC控制寄存器 0 (IICC0)的第 7 位(IICE0)为 1 实现)前，使用CLX0、SMC0、CL01 和 CL00 决定$I^2C$ 的传送时钟频率。若改变传送时钟频率，必须把IICE0 清零。**

备注　**1.** ×：　不必考虑
**2.** $f_{PRS}$：　外围硬件时钟频率
**3.** $f_{EXSCL0}$：　来自 EXSCL0 引脚的外部时钟频率

**(7) 端口模式寄存器 6 (PM6)**

此寄存器可以按位设置端口 6 的输入/输出操作模式。

当 P60/SCL0 引脚用于时钟 I/O 同时将 P61/SDA0 引脚用于串行数据 I/O 时，将 PM60 和 PM61 清零，且将 P60 和 P61 的输出锁存器清零。

在设置输出模式之前应将 IICE0 (IIC 控制寄存器 0 (IICC0)的第 7 位)设为 1，这是因为当 IICE0 为 0 时，P60/SCL0 和 P61/SDA0 引脚输出低电平(恒定)。

可由 1 位或 8 位存储器操作指令设置 PM6。

复位信号的产生将 PM6 设置为 FFH。

**图 16-10. 端口模式寄存器 6 (PM6) 的格式**

地址： FF26H　复位后： FFH　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM6 | PM67 | PM66 | PM65 | PM64 | PM63 | PM62 | PM61 | PM60 |

| PM6n | P6n 引脚 I/O 模式选择 (n = 0 ～ 7) |
|---|---|
| 0 | 输出模式 (输出缓冲器打开) |
| 1 | 输入模式 (输出缓冲器关闭) |

## 16.4 I²C 总线模式功能

### 16.4.1 引脚配置

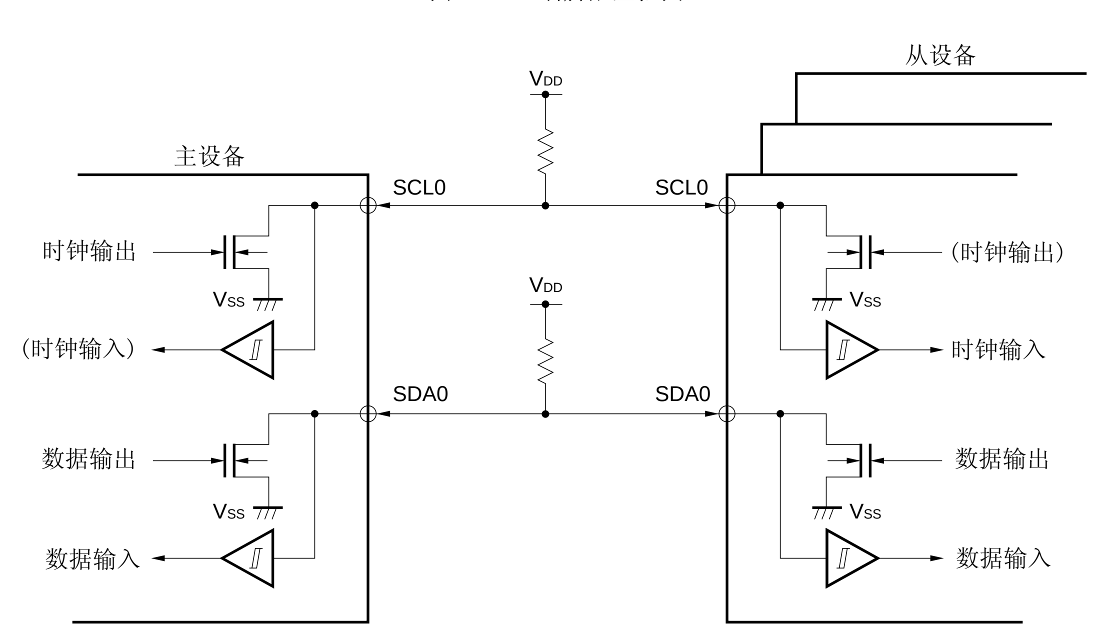
串行时钟引脚 (SCL0)和串行数据总线引脚(SDA0)配置如下。

(1) SCL0....... 此引脚用于串行时钟的输入和输出。
此引脚是主设备和从设备的 N-ch 漏极开路输出。该输入是施密特(Schmitt)输入。

(2) SDA0 ...... 此引脚用于串行数据的输入和输出。
此引脚是主设备和从设备的 N-ch 漏极开路输出。该输入是施密特(Schmitt)输入。

由于串行时钟线和串行数据总线的输出都是 N-ch 漏极开路输出，因此需要一个外部上拉电阻。

**图 16-11. 引脚配置框图**



## 16.5 I²C总线定义和控制方法

以下描述了I²C总线的串行数据通信格式和I²C总线使用的信号。 图 16-12 显示了通过I²C串行数据总线的“起始条件”， “地址”， “数据”和“停止条件”等信号输出时的传送时序。

**图 16-12. I²C总线串行数据传送时序**

SCL0　1-7　8　9　1-8　9　1-8　9

SDA0

起始条件　地址　R/$\overline{\text{W}}$　$\overline{\text{ACK}}$　数据　$\overline{\text{ACK}}$　数据　$\overline{\text{ACK}}$　停止条件

主设备产生起始条件、从设备地址和停止条件。

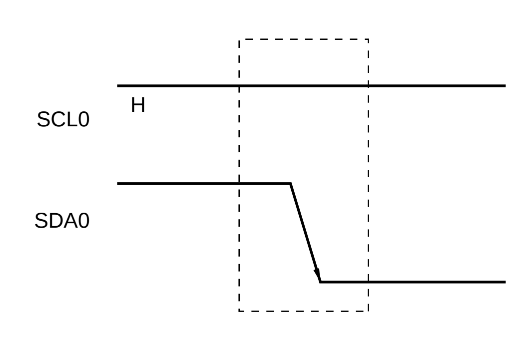
应答信号($\overline{\text{ACK}}$)可以由主设备或者从设备产生 (通常，由接收 8 位数据的设备输出)。

主设备持续输出串行时钟 (SCL0)。但在从设备中，可以延长 SCL0 的低电平阶段同时可以插入等待时间。

### 16.5.1 起始条件

SCL0 引脚处于高电平且 SDA0 引脚从高电平变为低电平时，满足起始条件。当启动串行传送时，SCL0 引脚和 SDA0 引脚的起始条件是由主设备产生并发给从设备的信号。当使用从设备时，可以检测到起始条件。

**图 16-13. 起始条件**




检测到停止条件(SPD0：IIC 状态寄存器 0 (IICS0)的第 0 位＝1)后，IIC 控制寄存器 0(IICC0)的第 1 位(STT0)被设置(为 1)时，输出一个起始条件。检测到起始条件时，IICS0 的第 1 位(STD0)被设置(为 1)。

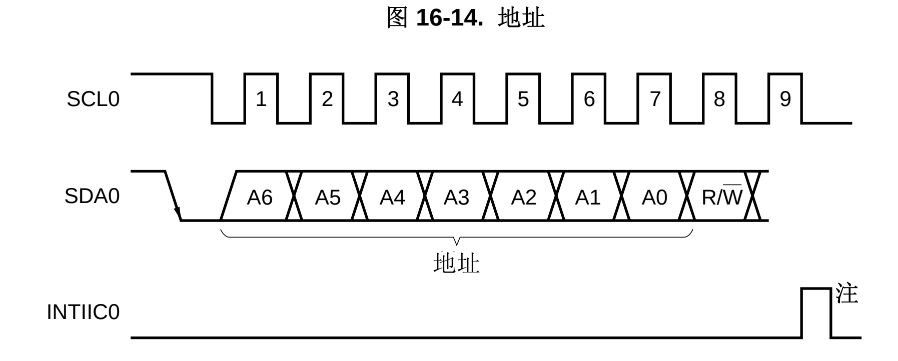
### 16.5.2 地址

地址由起始条件之后的 7 位数据指定。

一个地址是一个输出的 7 位数据段，用于选择一个通过总线与主设备相连的从设备。因此，通过总线相连的每个从设备必须有一个唯一的地址。

从设备包括具有以下功能的硬件：检测起始条件，检查 7 位地址数据是否与存储在从设备地址寄存器 0(SVA0)中的数据相等。如果地址数据和 SVA0 的值相等，则该从设备将被选中且与主设备通信，直到主设备产生一个起始条件或停止条件。

**图 16-14. 地址**




注　如果在从设备操作期间接收到的是数据而不是本地地址或扩展码时，则 INTIIC0 不会发出。

将从设备地址和第 8 位(用于指示传送方向(如 **16.5.3 传送方向指示**中所描述的))一起写入 IIC 移位寄存器 0(IIC0)并且输出。接收到的地址写入 IIC0。

将从设备地址分配给 IIC0 的高 7 位。

### 16.5.3 传送方向指示

除了 7-位地址数据之外，主设备还发送 1 位数据用于指示传送方向。

当传送方向指示位的值为“0”时，表示主设备正向从设备传送数据。当传送输方向指示位的值为“1”时，表示主设备正接收来自从设备的数据。

**图 16-15. 传送方向指示**

SCL0 1 2 3 4 5 6 7 8 9

SDA0 A6 A5 A4 A3 A2 A1 A0 R/W

传送方向指示

INTIIC0 注

注　如果在从设备操作期间接收到的是数据而不是本地地址或扩展码时，则 INTIIC0 不会发出。

### 16.5.4 应答信号 ($\overline{\text{ACK}}$)

$\overline{\text{ACK}}$ 信号用于检测处于发送方和接收方的串行数据的状态。

当接收方收到 8 位数据后将返回一个 $\overline{\text{ACK}}$ 信号。

发送方通常在发送了 8 位数据后接收到 $\overline{\text{ACK}}$ 信号。当 $\overline{\text{ACK}}$ 信号从接收方返回时，认为已正确接收并继续之后的处理。是否检测到 $\overline{\text{ACK}}$ 信号可以通过 IIC 状态寄存器 0(IICS0)的第 2 位(ACKD0)进行检测。

主设备接收到最后一个数据项时，不再返回 $\overline{\text{ACK}}$ 信号而是产生一个停止条件。如果从设备在接收到数据后未返回 $\overline{\text{ACK}}$ 信号，主设备将输出一个停止条件或重启条件并且停止传送。如果未返回 $\overline{\text{ACK}}$ 信号，可能是如下原因。

<1>　接收操作没有正常进行。

<2>　接收到最后一个数据项。

<3>　由地址指示的接收方不存在。

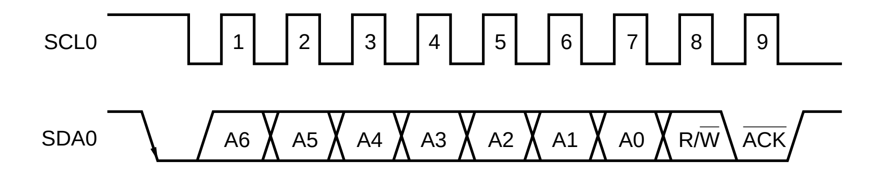
要产生 $\overline{\text{ACK}}$ 信号，接收方必须在第 9 个时钟时使 SDA0 线变为低电平(表示正常接收)。

可通过设置 IIC 控制寄存器 0(IICC0)的第 2 位(ACKE0)为 1，自动产生 $\overline{\text{ACK}}$ 信号。IICS0 寄存器的第 3 位(TRC0)可被数据的第 8 位(紧接着 7 位地址信息)设置。通常，设置 ACKE0=1 用于接收 (TRC0 = 0)。

如果一个从设备在接收(TRC0 = 0)期间不再收到任何数据或者不再需要下一个数据项，则从设备必须通过将 ACKE0 清零通知主设备，这样它就不会再收到任何数据。

接收(TRC0 = 0)期间，当主设备不需要下一个数据项时，必须将 ACKE0 清零，这样就不会产生 $\overline{\text{ACK}}$。在这种方式下，主设备通知作为发送方的从设备，不再需要其他数据(发送过程将停止)。

**图 16-16. $\overline{\text{ACK}}$ 信号**




接收到本地地址时，无论 ACKE0 的值是多少，都将自动产生 $\overline{\text{ACK}}$ 信号。当接收到一个非本地地址时，则不产生 $\overline{\text{ACK}}$ 信号(NACK)。

接收到扩展码时，如果 ACKE0 预先设为 1，则将产生 $\overline{\text{ACK}}$ 信号。

接收到数据时如何产生 $\overline{\text{ACK}}$ 信号，将根据等待时序的设置而变化。

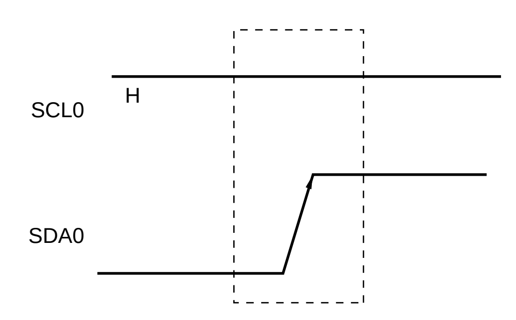
- 当选择 8 时钟等待状态时 (IICC0 寄存器的第 3 位(WTIM0)= 0)：
  退出等待状态前通过将 ACKE0 设为 1，在 SCL0 引脚第 8 个时钟的下降沿产生 $\overline{\text{ACK}}$ 信号。
- 当选择 9 时钟等待状态时 (IICC0 寄存器的第 3 位(WTIM0)= 1)：
  通过预先设置 ACKE0 为 1 产生 $\overline{\text{ACK}}$ 信号。

### 16.5.5 停止条件

SCL0 引脚处于高电平时，将 SDA0 引脚从低电平变为高电平可以产生一个停止条件。

停止条件是一个信号，它在串行传送操作结束时由主设备产生并发送给从设备。当作为从设备使用时，可以检测到停止条件。

**图 16-17. 停止条件**




IIC 控制寄存器 0(IICC0)的第 0 位(SPT0)设为 1 时，会产生一个停止条件。当检测到停止条件时，IIC 状态寄存器 0(IICS0)的第 0 位(SPD0)被设为 1 且当 IICC0 的第 4 位(SPIE0)被设为 1 时产生 INTIIC0。

### 16.5.6 等待

等待状态用来通知通信另一方，设备(主设备或从设备)正准备发送或者接收数据(即处于等待状态)。

设置 SCL0 引脚为低电平，通知通信另一方设备已进入等待状态。当主设备和从设备的等待状态被取消时，下一个数据传送才能开始。

图 16-18. 等待 (1/2)

**(1)　当主设备输出第 9 个时钟后等待而从设备输出第 8 个时钟后等待时(主设备发送，从设备接收，同时 ACKE0 = 1)**

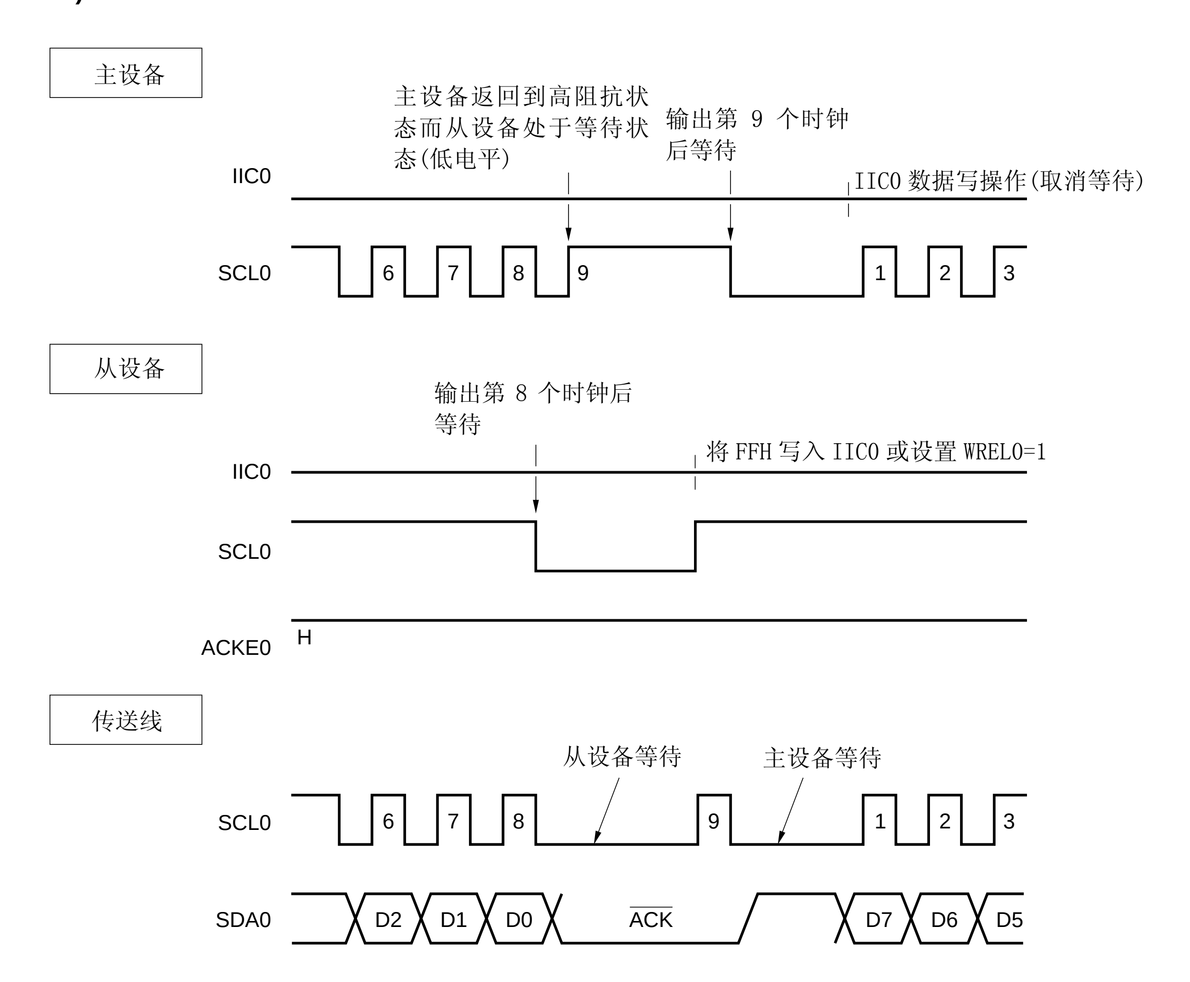
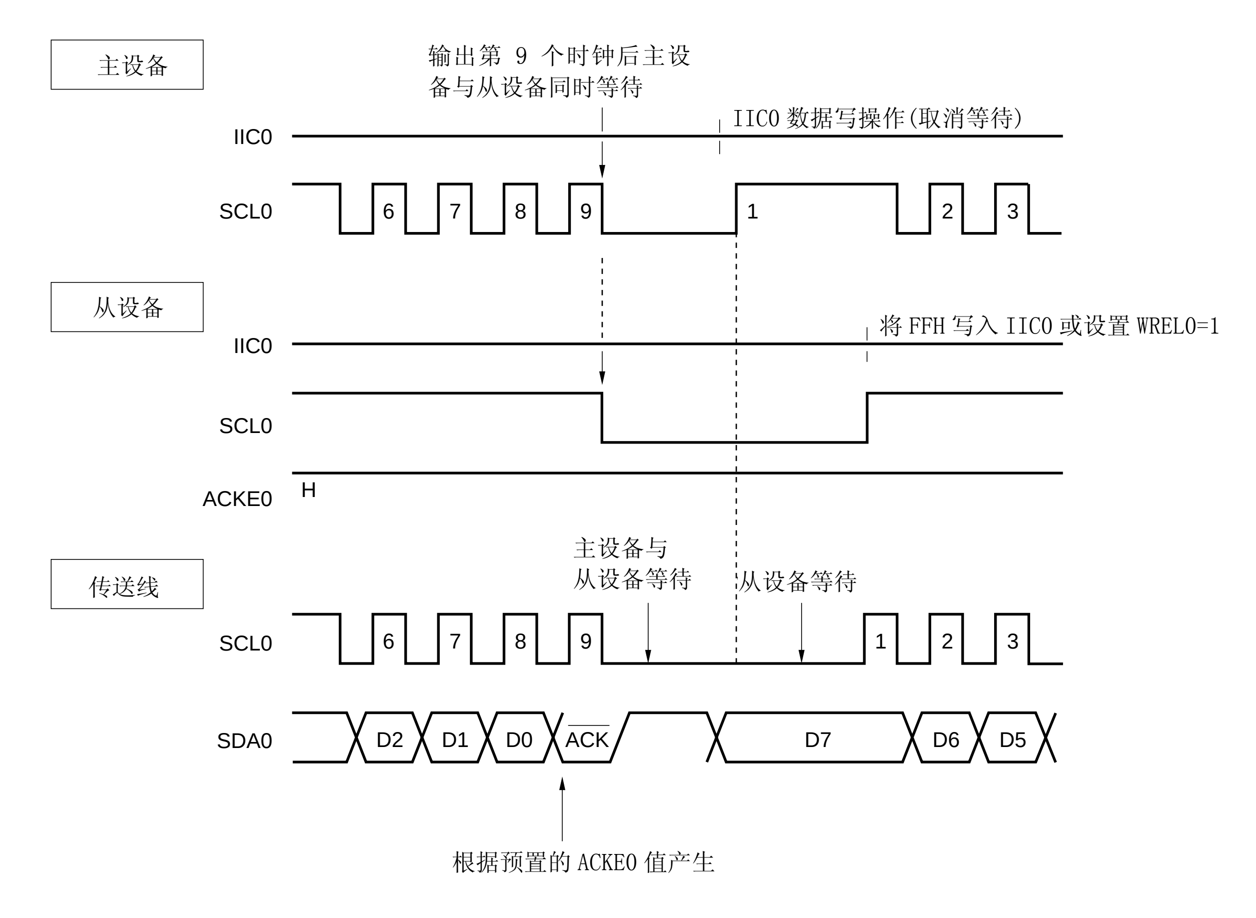


图 **16-18.** 等待 **(2/2)**

**(2)** 当主和从设备都输出第 **9** 个时钟后等待时**(**主设备发送，从设备接收，同时 **ACKE0 = 1)**




备注　ACKE0: IIC 控制寄存器 0(IICC0)的第 2 位
　　　WREL0: IIC 控制寄存器 0(IICC0)的第 5 位

根据 IIC 控制寄存器 0(IICC0)的第 3 位(WTIM0)的设置，自动产生等待状态。

通常，当 IICC0 的第 5 位(WREL0)设置为 1 时或者在将 FFH 写入 IIC 移位寄存器 0(IIC0)时，接收方取消等待状态，而当数据写入 IIC0 时，发送方取消等待状态。

主设备也能通过以下任何一种方法取消等待状态。

- 通过把 IICC0 的第 1 位(STT0)设为 1
- 通过把 IICC0 的第 0 位(SPT0)设为 1

### 16.5.7 取消等待

$I^2C$通常通过如下处理方式取消一个等待状态。

- 把数据写入 IIC 移位寄存器 0(IIC0)
- 设置 IIC 控制寄存器 0(IICC0) 的第 5 位(WREL0) (取消等待)
- 设置 IICC0 寄存器的第 1 位 (STT0) (产生起始条件)[注]
- 设置IICC0 寄存器的第 0 位 (SPT0) (产生停止条件)[注]

**注** 仅用于主设备

当执行以上取消等待处理过程时，$I^2C$ 取消等待状态且恢复通信。

通过向 IIC0 写数据来取消等待状态并发送数据 (包括地址)。

通过设置 IIC 控制寄存器 0(IICC0)的第 5 位(WREL0)为 1，可在取消等待状态之后接收数据，或完成数据发送操作。

设置 IICC0 的第 1 位(STT0)为 1，可在取消一个等待状态之后产生一个重启条件。

设置 IICC0 的第 0 位(SPT0)为 1，可在取消一个等待状态之后产生一个停止条件。

对于一个等待状态只能执行一次取消操作。

例如，如果通过设置 WREL0 为 1 在取消等待状态之后将数据写入 IIC0，则可能一个不正确的值输出到 SDA0，这是因为改变 SDA0 线的时序和 IIC0 写操作时序相冲突。

除此以外，如果通信已放弃时 IICE0 被清零，则停止通信，这样可以取消等待状态。

如果$I^2C$ 总线由于噪音产生死锁，则通过设置IICC0 的第 6 位(LREL0)可以保存通信处理内容，这样等待状态便可以取消。

### 16.5.8 中断请求 (INTIIC0)产生时序和等待控制

IIC 控制寄存器 0(IICC0)的第 3 位(WTIM0)的设置决定 INTIIC0 产生时序和相应的等待控制，如表 18-3 所示。

**表 16-3. INTIIC0 产生时序和等待控制**

| WTIM0 | 从设备操作期间 | | | 主设备操作期间 | | |
|---|---|---|---|---|---|---|
| | 地址 | 数据接收 | 数据发送 | 地址 | 数据接收 | 数据发送 |
| 0 | 9[注 1, 2] | 8[注 2] | 8[注 2] | 9 | 8 | 8 |
| 1 | 9[注 1, 2] | 9[注 2] | 9[注 2] | 9 | 9 | 9 |

**注** 1. 仅当有地址与从设备地址寄存器 0(SVA0)设置的地址相等时，在第 9 个时钟的下降沿产生从设备的 INTIIC0 信号和等待周期。

此时，不论 IICC0 的第 2 位(ACKE0)设置为何值，都将产生 $\overline{ACK}$ 信号。对于一个接收到扩展码的从设备，在第 8 个时钟的下降沿将产生 INTIIC0。

但是，如果重启之后地址不相等，将在第 9 个时钟的下降沿产生 INTIIC0，但不会产生等待周期。

2. 如果接收地址与从设备地址寄存器 0(SVA0)的内容不相等，同时也未接收到扩展码，则 INTIIC0 和等待周期都不会产生。

**备注** 表中的数字是指串行时钟的时钟信号个数。中断请求和等待控制和这些时钟信号的下降沿同步。

**(1) 地址发送/接收期间**

- 从设备操作: 中断和等待时序取决于以上注 1、2 中所描述的情况，与 WTIM0 位无关。
- 主设备操作: 中断和等待时序在第 9 个时钟的下降沿发生，与 WTIM0 位无关。

**(2) 数据接收期间**

- 主/从设备操作: 中断和等待时序取决于 WTIM0 位。

**(3) 数据发送期间**

- 主/从设备操作: 中断和等待时序取决于 WTIM0 位。

**(4) 等待取消的方法**

四种取消等待方法如下所示。

- 向 IIC 移位寄存器 0 (IIC0)写入数据
- 设置 IIC 控制寄存器 0(IICC0)的第 5 位 (WREL0) (取消等待)
- 设置IICC0 寄存器的第 1 位 (STT0) (产生起始条件)[注]
- 设置IICC0 寄存器的第 0 位 (SPT0) (产生停止条件)[注]

**注** 仅用于主设备。

当选择了一个 8-时钟等待时(WTIM0 = 0)，必须先决定是否产生 $\overline{\text{ACK}}$ 信号，再考虑取消等待。

**(5) 停止条件的检测**

当检测到一个停止条件时(仅当 SPIE0 = 1)，INTIIC0 将会产生。

### 16.5.9 地址相等的检测方法

在$I^2C$ 总线模式下，主设备可以通过发送的从设备地址来选择指定的从设备。

可以通过硬件自动检测到地址相等。当将一个本地地址设置到从设备地址寄存器 0(SVA0)中且该地址与主设备发送的从设备地址相等时，或者接收到扩展码时，都将产生一个中断请求(INTIIC0)。

### 16.5.10 错误检测

$I^2C$总线模式下，在数据发送期间通过发送设备的IIC移位寄存器(IIC0)捕获串行数据总线(SDA0) 的状态，因此发送前的IIC0 数据可以用来和已发送的IIC0 数据比较，以此来实现对发送错误的检测。当比较值不相等时，可以判断存在一个发送错误。

### 16.5.11 扩展码

(1) 当接收地址的高 4 位为 “0000”或“1111”时，代表接收到扩展码的扩展码接收标志(EXC0)被设为 1，并且在第 8 个时钟的下降沿产生中断请求(INTIIC0)。而存放在从设备地址寄存器 0(SVA0)中的本地地址不受影响。

(2) 如果由一个 10 位地址传送设置 SVA0 为“11110××0”且主设备传送的值为“11110××0”，则其结果如下所示。注意在第 8 个时钟的下降沿产生 INTIIC0。

- 数据的高 4 位匹配:　　　　　EXC0 = 1
- 7 位数据匹配:　　　　　　　COI0 = 1

备注　EXC0:  IIC 状态寄存器 0(IICS0)的第 5 位
　　　COI0:  IIC 状态寄存器 0(IICS0)的第 4 位

(3) 由于中断请求发生之后的处理过程随扩展码之后的数据而有所不同，因此这样的处理过程由软件来执行。
如果当一个从设备正在操作时接收到扩展码，那么该从设备(即使它的地址不匹配)也将参与通信。
例如，接收到扩展码后，如果不希望目标设备作为从设备来操作，则将 IIC 控制寄存器 0(IICC0)的第 6 位(LREL0)设为 1，为下一个通信操作设置待机模式。

**表 16-4. 扩展码位定义**

| 从设备地址 | R/W 位 | 描述 |
|---|---|---|
| 0 0 0 0 0 0 0 | 0 | 通用调用地址 |
| 0 0 0 0 0 0 0 | 1 | 起始字节 |
| 0 0 0 0 0 0 1 | × | C-总线地址 |
| 0 0 0 0 0 1 0 | × | 为不同总线格式预约的地址 |
| 1 1 1 1 0 X X | × | 10 位从设备地址规范 |

### 16.5.12 仲裁

当多个主设备同时产生起始条件信号时 (在 STD0 设为 1 之前将 STT0 设为 1)，调整时钟数目直至传送数据改变，然后执行主设备间的通信。这种操作称为仲裁。

当有一个主设备仲裁失败时，可通过仲裁失败产生的时序设置 IIC 状态寄存器 0(IICS0)中的仲裁失败标志(ALD0)为 1，同时 SCL0 和 SDA0 线变为高阻抗，从而释放总线。

基于下一个中断请求的时序(第 8 或第 9 个时钟，检测到停止条件时等)以及由软件设置 ALD0=1 来检测仲裁失败。

有关中断请求时序，详见 **16.5.17 I²C 中断请求(INTIIC0)**产生时序。

**备注**　STD0: IIC 状态寄存器 0 (IICS0)的第 1 位
　　　　STT0: IIC 控制寄存器 0 (IICC0)的第 1 位

**图 16-19. 仲裁时序举例**

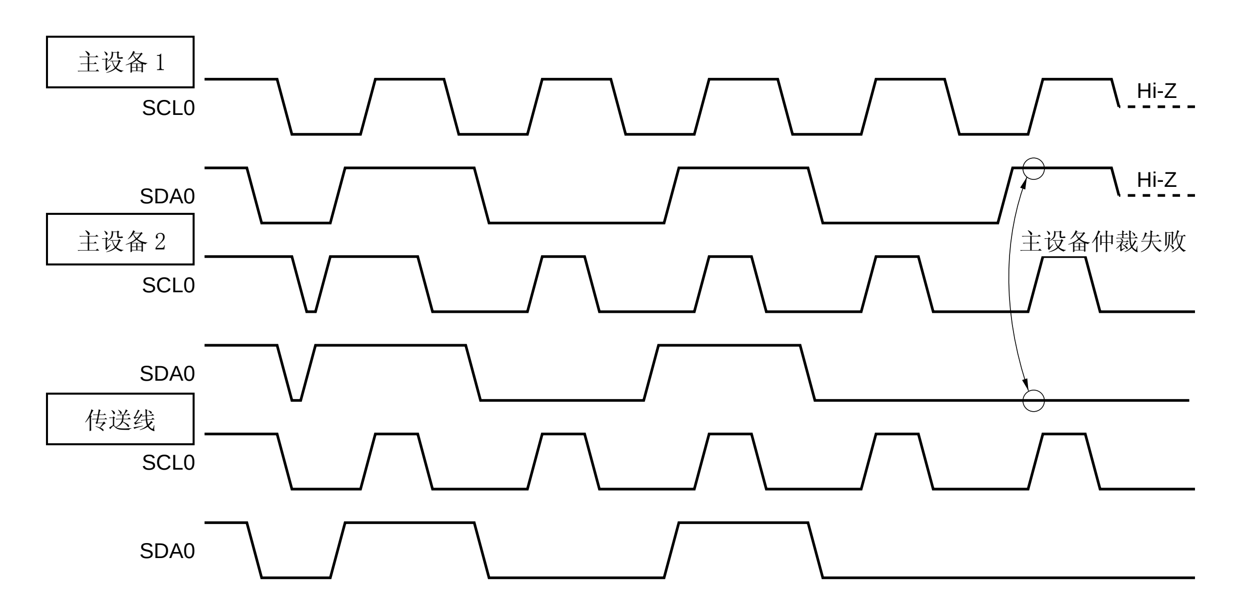


**表 16-5. 仲裁期间的状态和中断请求产生时序**

| 仲裁期间的状态 | 中断请求产生时序 |
| --- | --- |
| 地址发送期间 | 字节传送后第 8 或第 9 个时钟的下降沿[注1] |
| 地址发送后读/写数据 | |
| 扩展码发送期间 | |
| 扩展码发送后读/写数据 | |
| 数据发送期间 | |
| 数据发送之后的 $\overline{\text{ACK}}$ 信号传送期间 | |
| 数据传送期间检测到重启条件时 | |
| 数据传送期间检测到停止条件时 | 停止条件产生时 (当SPIE0 = 1)[注2] |
| 准备产生重启条件时数据为低电平 | 字节传送后第 8 或第 9 个时钟的下降沿[注1] |
| 准备产生重启条件时检测到停止条件 | 停止条件产生时 (当SPIE0 = 1)[注2] |
| 准备产生停止条件时数据为低电平 | 字节传送后第 8 或第 9 个时钟的下降沿[注1] |
| 准备产生重启条件时 SCL0 为低电平 | |

注 **1.** 当 WTIM0 (IIC 控制寄存器 0 (IICC0)的第 3 位) = 1 时,将在第 9 个时钟的下降沿产生一个中断请求。当 WTIM0 = 0 且扩展码的从设备地址已被接收时，将在第 8 个时钟的下降沿产生一个中断请求。

**2.** 如果可以产生仲裁，则设置 SPIE0 = 1，进行主设备操作。

**备注** SPIE0: IIC 控制寄存器 0 (IICC0)的第 4 位

### 16.5.13 唤醒功能

$I^2C$总线从设备唤醒功能是，在接收到本地地址及扩展码时产生一个中断请求信号(INTIIC0)。

此功能通过在地址不相等时禁止产生不必要的 INTIIC0 信号，使处理更加有效。

当检测到一个起始条件时，设置唤醒待机模式。仲裁失败可能将主设备(已产生启动条件)转变为从设备，如果因此而发送地址时，则该唤醒待机模式有效。

可是，当检测到停止条件时，无论是否有唤醒功能，都设置 IIC 控制寄存器 0 (IICC0) 的第 4 位(SPIE0)，这将决定是否允许或禁止中断请求。

### 16.5.14 通信预约

**(1) 允许使用通信预约功能时(IIC 标志寄存器 0(IICF0)的第 0 位(IICRSV) = 0)**

如果要启动当前不使用总线的主设备通信，可以使用通信预约功能，在总线释放时允许发送一个启动条件信号。有两种不使用总线时的模式。

- 当仲裁结果不是主/从设备操作时
- 当接收到扩展码且禁止从设备操作时(不返回 $\overline{\text{ACK}}$ 信号且 IIC 控制寄存器 0 (IICC0) 的第 6 位(LREL0)被设置为 1 时释放总线)。

如果总线未使用时(检测到停止条件后)IICC0 的第 1 位(STT0) 被设为 1，则自动产生一个起始条件并且设置等待状态。
如果在 IICC0 的第 4 位(SPIE0)被设为 1 之后将地址写入 IIC 移位寄存器 0 (IIC0)，并且通过总线释放(检测停止状态)产生中断请求信号(INTIIC0)来检测该地址，则设备将以主设备方式自动开始通信。在检测到停止条件前写入 IIC0 的数据是无效的。
当 STT0 已被设为 1 时，操作模式(作为起始条件或作为通信预约)取决于总线状态。

- 如果总线已被释放........................................................ 产生一个起始条件
- 如果总线未被释放 (待机模式) ...................................... 通信预约

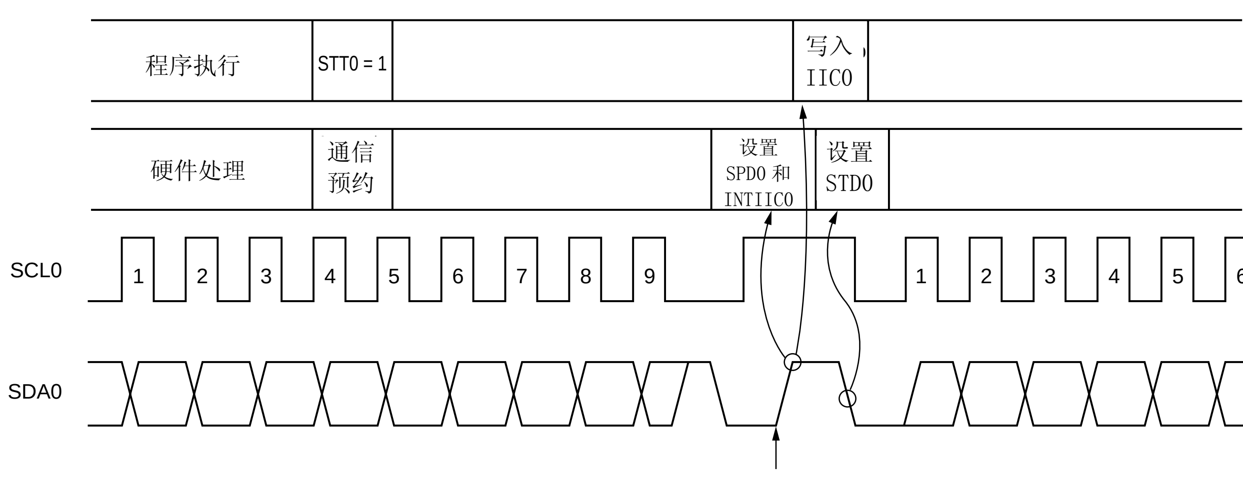
STT0 设为 1 且经历了等待时间后，使用 MSTS0(IIC 状态寄存器 0 (IICS0)的第 7 位) 检查是否进行通信预约操作。
应通过软件设置的等待周期见表 16-6。

★ **表 16-6. 等待周期**

| CLX0 | SMC0 | CL01 | CL00 | 等待周期 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 46 个时钟 |
| 0 | 0 | 0 | 1 | 86 个时钟 |
| 0 | 0 | 1 | 0 | 172 个时钟 |
| 0 | 0 | 1 | 1 | 34 个时钟 |
| 0 | 1 | 0 | 0 | 30 个时钟 |
| 0 | 1 | 0 | 1 | |
| 0 | 1 | 1 | 0 | 60 个时钟 |
| 0 | 1 | 1 | 1 | 12 个时钟 |
| 1 | 1 | 0 | 0 | 18 个时钟 |
| 1 | 1 | 0 | 1 | |
| 1 | 1 | 1 | 0 | 36 个时钟 |

图 16-20 显示通信预约时序。

**图 16-20. 通信预约时序**




备注　IIC0:　IIC 移位寄存器 0
　　　STT0:　IIC 控制寄存器 0 (IICC0)的第 1 位
　　　STD0:　IIC 状态寄存器 0 (IICS0)的第 1 位
　　　SPD0:　IIC 状态寄存器 0 (IICS0)的第 0 位

通过以下时序接受通信预约。IIC 状态寄存器 0 (IICS0)的第 1 位(STD0)设为 1 之后，可以通过在检测到停止条件之前设置 IIC 控制寄存器 0 (IICC0)的第 1 位(STT0)为 1 来进行通信预约。

**图 16-21. 接受通信预约的时序**

SCL0

SDA0

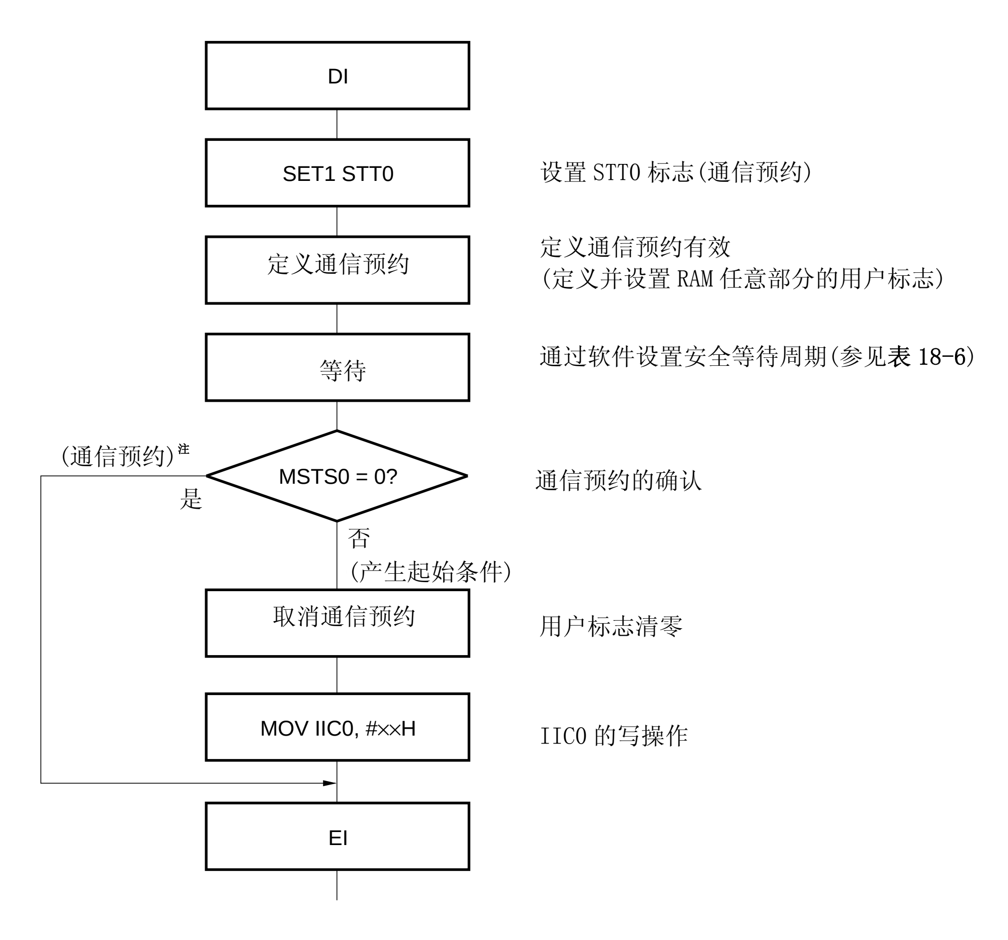
STD0

SPD0

待机模式

图 16-22 显示通信预约方案。

图 **16-22.** 通信预约方案




**注**　当产生一个停止条件中断请求时，通信预约执行对 IIC 移位寄存器 0 (IIC0)的写操作。

**备注**　STT0:　IIC 控制寄存器 0 (IICC0)的第 1 位
　　　MSTS0: IIC 状态寄存器 0 (IICS0)的第 7 位
　　　IIC0:　IIC 移位寄存器 0

**(2) 当禁止使用通信预约功能时 (IIC 标志寄存器 0 (IICF0)的第 0 位(IICRSV) = 1)**

如果在总线通信期间，总线未使用时将 IIC 控制寄存器 0 (IICC0) 的第 1 位(STT0)设置为 1，则该请求会被拒绝并且不产生起始条件。下列两种状态包括了总线未使用时的情况。

- 当仲裁结果不是主/从设备操作时
- 接收到一个扩展码且禁止从设备操作时 (不返回 $\overline{\text{ACK}}$ 信号且 IIC 控制寄存器 0 (IICC0) 的第 6 位(LREL0)被设置为 1 时释放总线)

通过检查 STCF(IICF0 的第 7 位)来确认是产生了起始条件还是请求被拒绝。设置 STT0 = 1 后，直到 STCF 被设置为 1 时，必须满足表 16-7 所示的时间。因此，必须通过软件确保等待时间。

表 **16-7.** 等待周期

| CL01 | CL00 | 等待周期 |
|---|---|---|
| 0 | 0 | 6 个时钟 |
| 0 | 1 | 6 个时钟 |
| 1 | 0 | 12 个时钟 |
| 1 | 1 | 3 个时钟 |

### 16.5.15 其他注意事项

(1) 当 STCEN (IIC 标志寄存器 0 (IICF0)的第 1 位) = 0 时

在允许$I^2C$操作(IICE0 = 1)后，无论总线的实际状态如何都可以识别总线通信状态(IICBSY (IICF0 的第 6 位) = 1)。从检测到非停止条件模式变为主设备通信模式时，首先产生一个停止条件以释放总线，然后进行主设备通信。

当使用多个主设备时，如果总线未被释放(没有检测到停止条件)，则不能进行主设备通信。

按下列步骤产生停止条件。

<1> 设置 IIC 时钟选择寄存器 0 (IICCL0)。
<2> 设置 IIC 控制寄存器 0 (IICC0) 的第 7 位(IICE0)为 1。
<3> 设置 IICC0 的第 0 位(SPT0)为 1。

(2) 当 STCEN = 1

在允许$I^2C$操作(IICE0 = 1)后，无论总线的实际状态如何都可以识别总线通信状态(IICBSY (IICF0 的第 6 位) = 1)。要产生第一个起始条件 (STT0 (IIC控制寄存器 0 (IICC0)的第 1 位) = 1), 必须确认总线已被释放，以免干扰其他通信。

(3) 如果其他$I^2C$ 通信已经进行

如果SDA0 引脚为低电平同时SCL0 引脚为高电平时允许$I^2C$ 操作，且设备已进行通信，则$I^2C$ 的宏可以识别SDA0 引脚已经变为低电平(检测一个起始条件)。如果此时总线上的值被识别为一个扩展码，则返回$\overline{ACK}$信号，但这会干扰其他$I^2C$ 通信。为了避免这种情况，可按下列步骤启动$I^2C$。

<1> 当检测到停止条件时，将 IICC0 的第 4 位(SPIE0)清零，禁止产生中断请求信号(INTIIC0)。
<2> 设置IICC0 的第 7 位(IICE0)为 1，允许$I^2C$的操作。
<3> 等待对启动条件的检测。
<4> $\overline{ACK}$ 信号返回之前(设置 IICE0 为 1 之后的 4 到 80 个时钟)设置 IICC 的第 6 位 (LREL0) 为 1，强行禁止检测。

(4) 允许操作之前(IICE0 = 1)通过使用 SMC0, CL01, CL00 (IICL0 的第 3，1 和 0 位)和 CLX0 (IICX0 的第 0 位)决定传送时钟频率。要改变传送时钟频率，必须清零 IICE0。

(5) 在对 STT0 和 SPT0 (IICC0 的第 1 和第 0 位)设置之后和清零之前，禁止再次设置。

(6) 当预约发送时，设置 SPIE0 (IICL0 的第 4 位)为 1，以允许在检测到停止条件时产生中断请求。中断请求产生后，将通信数据写入 IIC0，传送过程开始。如果检测到停止条件时不产生中断请求，设备将处于等待状态，因为开始通信时不产生中断请求。但是，由软件检测 MSTS0 (IICS0 的第 7 位)时不必设置 SPIE0 为 1。

### 16.5.16 通讯操作

以下所示三种操作的流程图。

<R> **(1) 主设备操作 (单主设备系统)**

在单主设备系统下使用 78K0/KC2 作为主控制器的流程图如下。

流程图分为初始设定和通讯处理。在开始时执行初始设定。如果要求与从设备通讯，则准备通讯并执行通讯处理。

<R> **(2) 主设备操作 (多主设备系统)**

在I$^2$C总线多主设备系统中，当总线参与通讯时，则不能通过I$^2$C总线规范来判断总线是否被释放还是在使用。这里，当数据和时钟在一定时期内（一帧）处于高电平，78K0/KC2 将在总线释放状态下参与通讯。

流程图分为初始设定、通讯等待和通讯处理。这里没有显示当 78K0/KE2 仲裁失败而作为从设备的处理，只显示主设备的处理。在启动时执行初始化设置参与通信。然后，主设备等待通信请求或者从设备等待详细指定。在通信处理中执行实际通信，从设备支持发送/接收，主设备之间进行仲裁。

<R> **(3) 从设备操作**

以下举例当 78K0/KC2 用做I$^2$C总线的从设备。

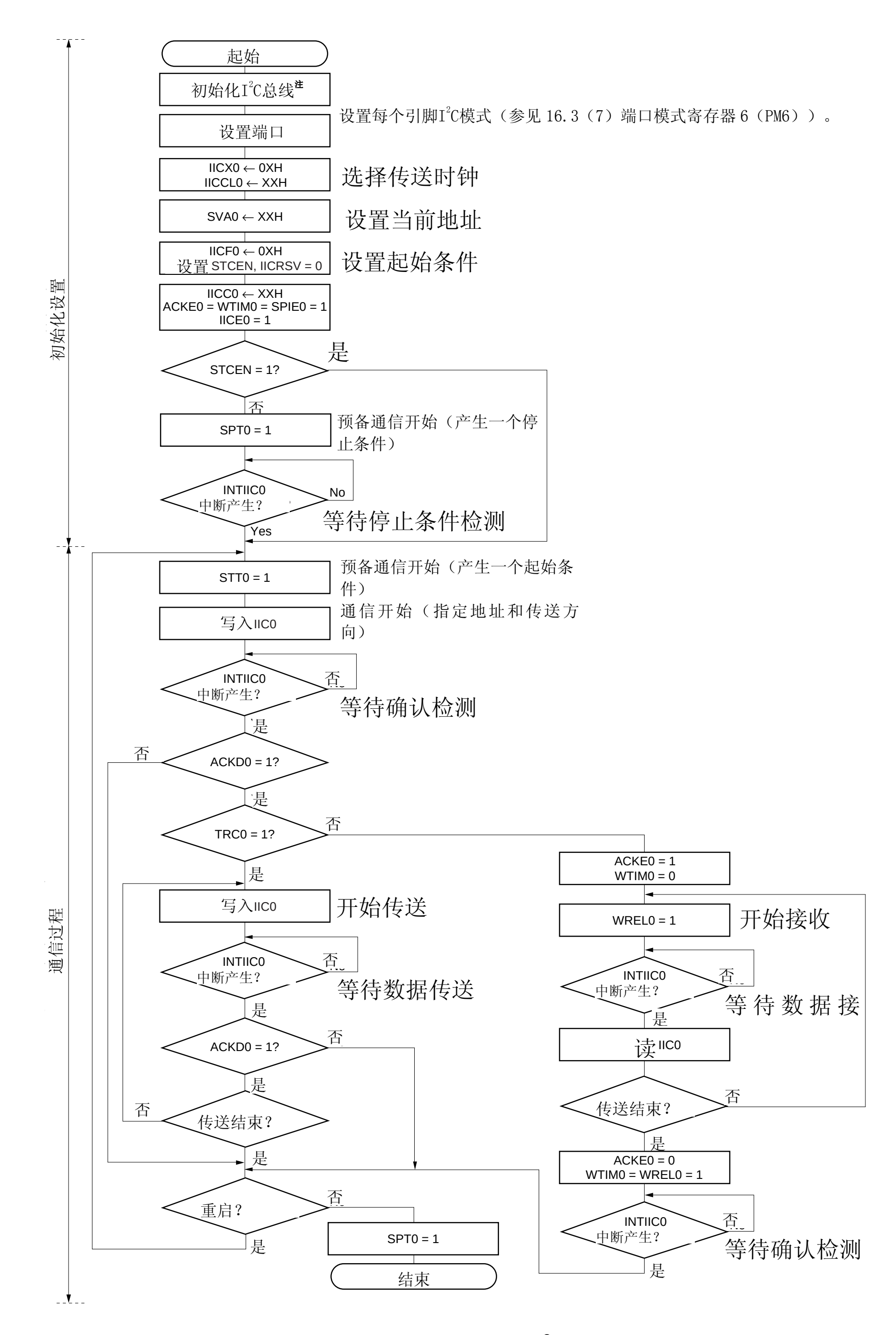
当作为从设备使用时，通过中断开始工作。在开始时执行初始化设定，然后等待 INTIIC0 中断产生（通讯等待）。当 INTIIC0 中断产生时，判定通讯状态并将结果作为标志传递给主程序。

通过验证这个标志位，实现必要的通讯处理。

(1) 主设备操作 (单主设备系统)

图 16-23. 主设备操作流程图 (单主设备系统)

<R>

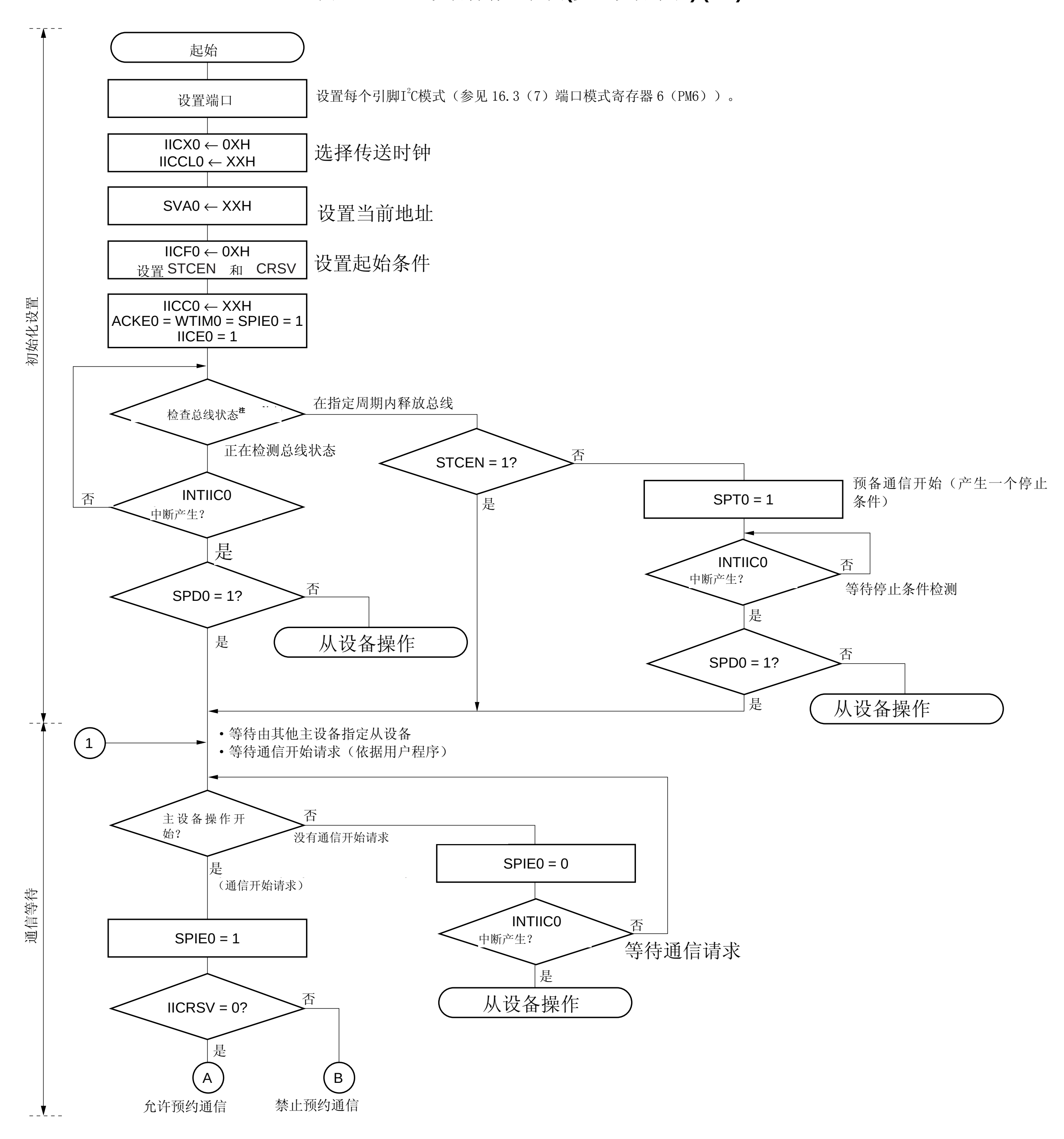



注 根据正在通信的产品规范释放(SCL0 和SDA0 引脚 = 高电平)I²C总线。如果EEPROM向SDA0 引脚输出一个低电平，例如，设SCL0 引脚为输出端口模式，且从输出端口输出一个时钟脉冲直到SDA0 引脚持续处于高电平。

**备注** 遵照正在通信的产品规范，注意发送和接收的格式。

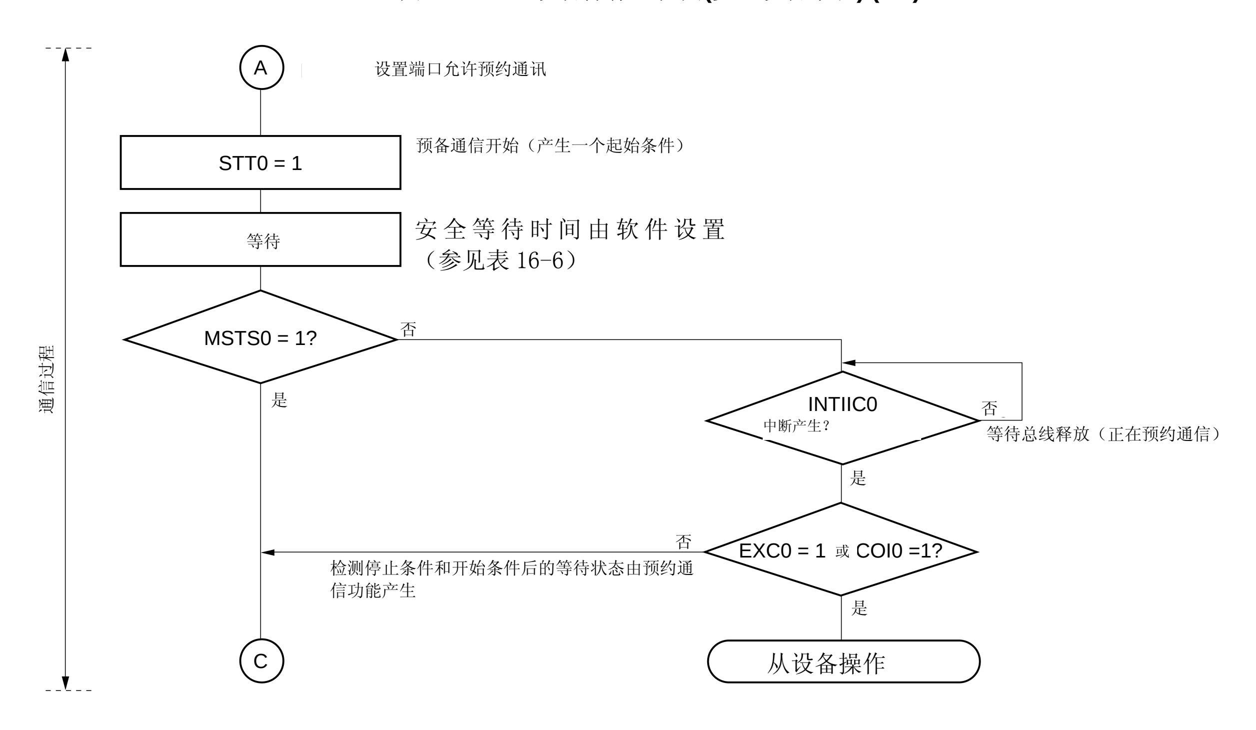
**(2) 主设备操作 (多主设备系统)**

**图 16-24. 主设备操作流程图(多主设备系统) (1/3)**




注　确认总线已释放了(CLD0 位 = 1, DAD0 位 = 1)指定的一段时间(如，一帧的周期)。如果SDA0 引脚持续处于低电平，可以根据正在使用的产品规范决定是否释放(SCL0 和 SDA0 引脚 = 高电平)I²C总线。

**图 16-24. 主设备操作流程图(多主设备系统) (2/3)**



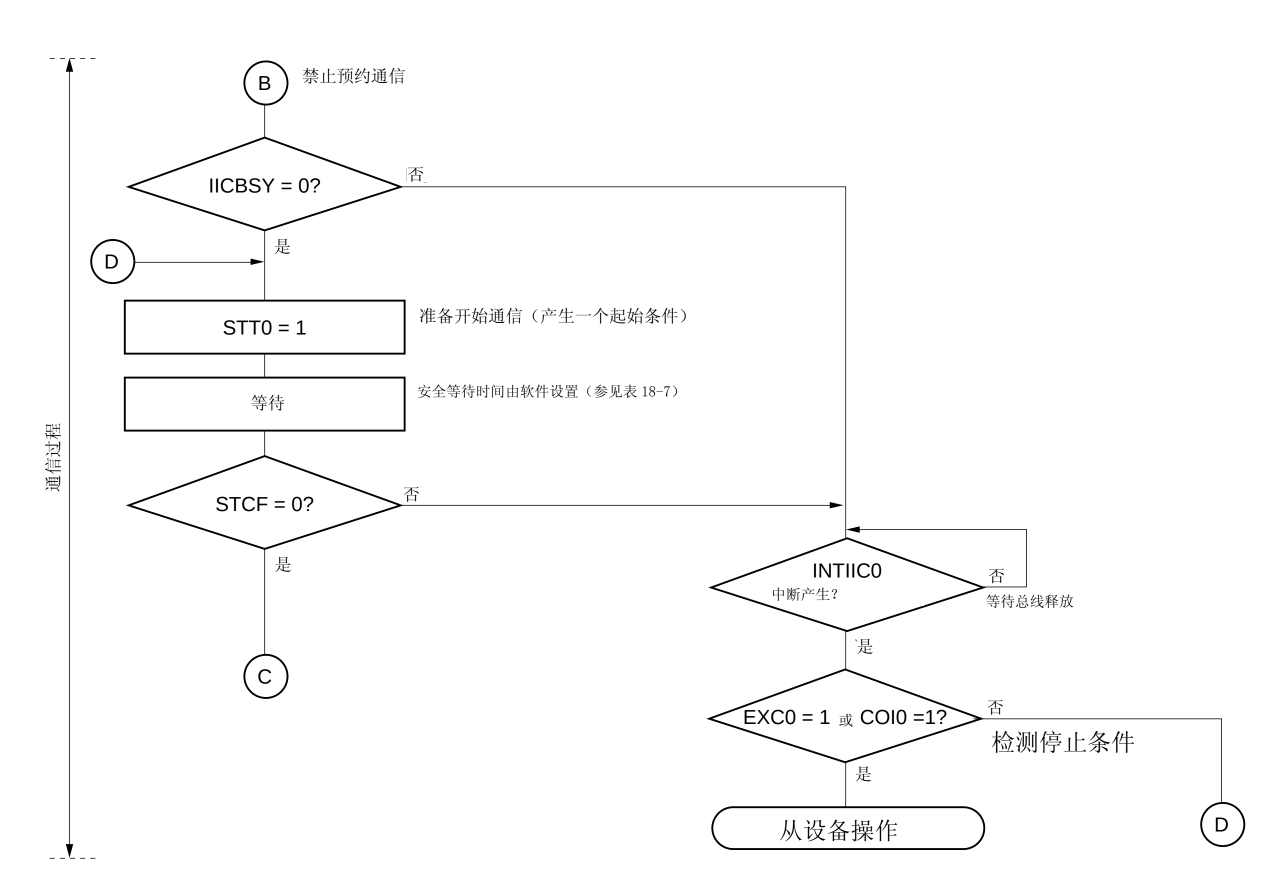
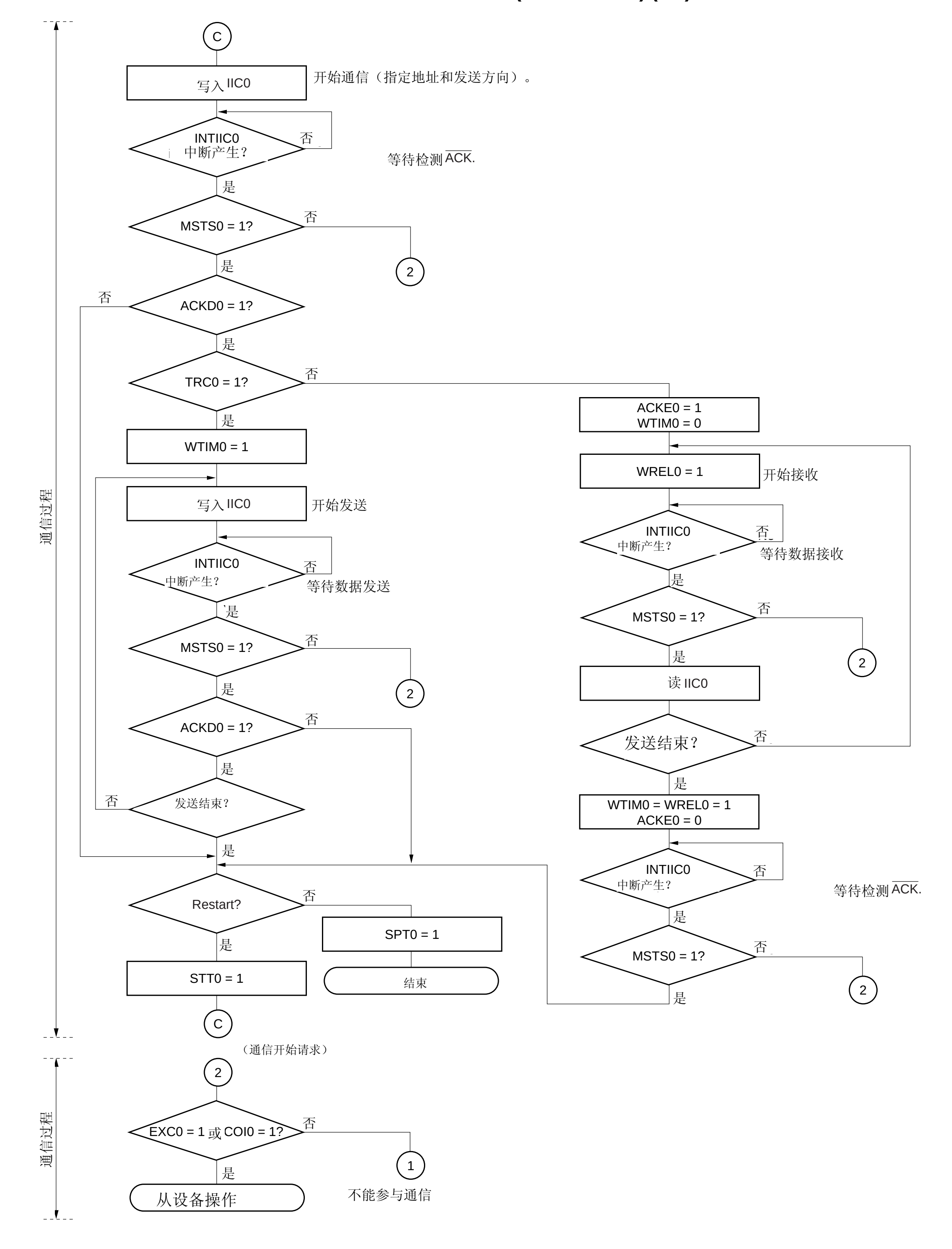



图 16-24. 主设备操作流程图(多主设备系统) (3/3)



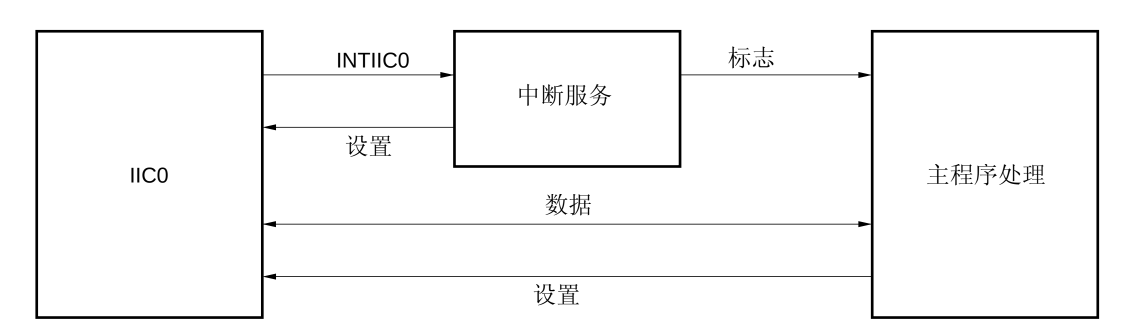

**备注** 
1. 遵照正在通信的产品规范，注意发送和接收的格式。
2. 如果在多主设备系统中作为一个主设备使用时，则应在每次中断 INTIIC0 发生后读取 MSTS0 位来检查仲裁结果。
3. 如果在多主设备系统中作为一个从设备使用时，则通过在每次中断 INTIIC0 发生后使用 IICS0 和 IICF0 寄存器来检查状态，并决定下一步的处理。

### (3) 从设备操作

从设备操作过程如下所示。

基本上，从设备操作是由事件驱动的。因此，必须执行 INTIIC0 中断处理过程(该中断处理过程必须完全改变操作状态例如通信时的停止条件检测)。

在下列说明中，假定数据通信不支持扩展码。同时假定 INTIIC0 中断服务只执行状态转换处理过程，且实际数据通信过程由主程序执行。




因此，设置以下 3 个标志，并传递给主程序而不是 INTIIC0，这样可以执行数据通信处理过程。

**<1> 通信模式标志**

此标志指示下列两种通信状态。

- 清除模式: 未进行数据通信
- 通信模式: 进行数据通信(从有效地址检测到停止条件检测，不检测来自主设备的 ACK 信号，地址不相等)

**<2> 准备标志**

此标志表示允许进行数据通信。它的功能和普通数据通信的 INTIIC0 中断相同。通过中断服务可以设置此标志，通过主程序可以将该标志清零。通信启动时，可由中断服务将此标志清零。但是，在发送第一个数据时不能由中断服务设置准备标志。因此，发送第一个数据时该标志未清零 (认为地址匹配是下一个数据发送请求)。

**<3> 通信方向标志**

此标志指示通信方向。它的值与 TRC0 相同。

从设备操作的主程序处理过程说明如下。

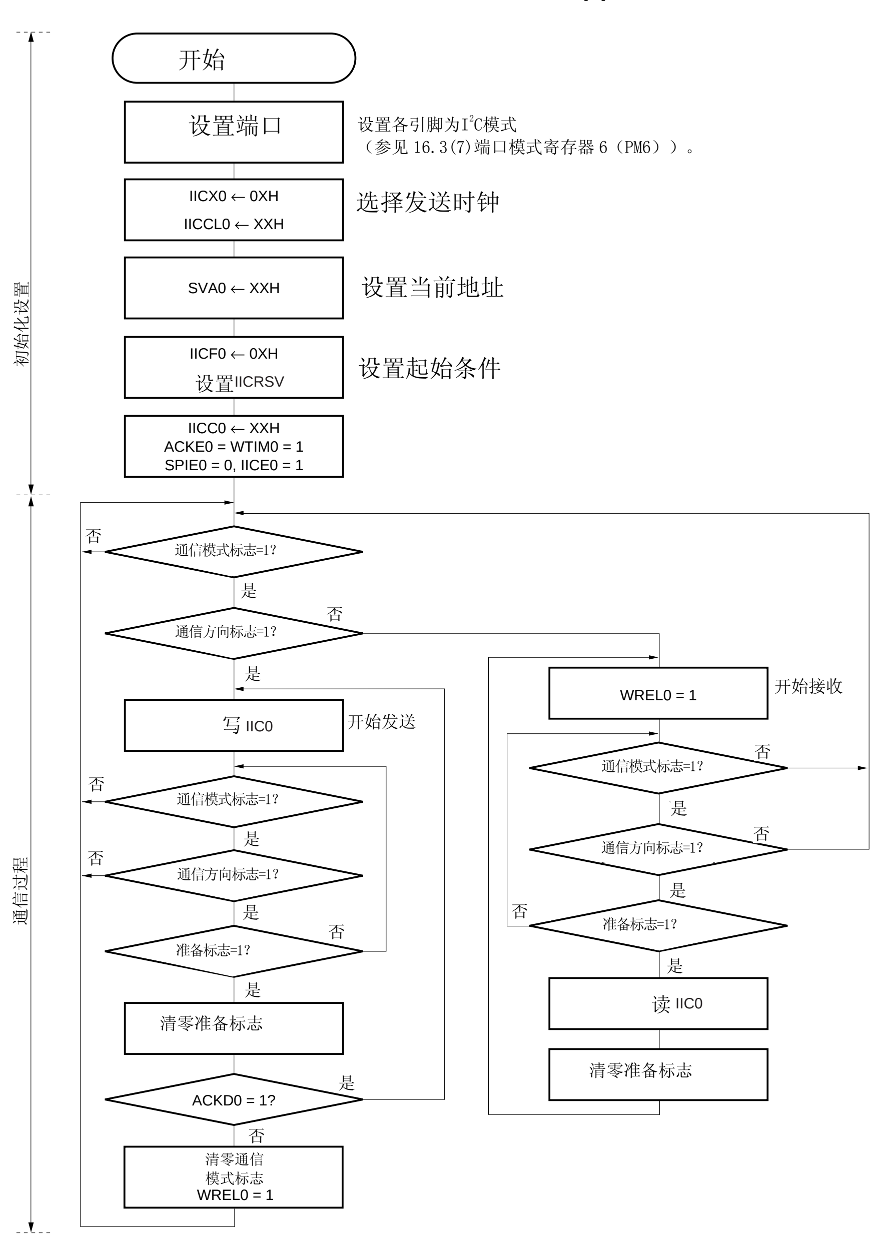
启动串行接口 IIC0 且等待直到允许通信。当允许通信时，通过使用通信模式标志和准备标志来执行通信 (由中断来执行停止条件和启动条件的处理。这里，通过标志来检查状态)。

重复发送操作直到主设备不再返回 $\overline{ACK}$ 信号。如果没有从主设备返回 $\overline{ACK}$ 信号，则通信结束。

对于接收操作，则要接收一定量的数据。通信完成时，$\overline{ACK}$ 信号不作为下一个数据返回。此后，主设备产生一个停止条件或重启条件。使用这种方法可从通信状态退出。

**图 16-25. 从设备操作流程图 (1)**

<R>



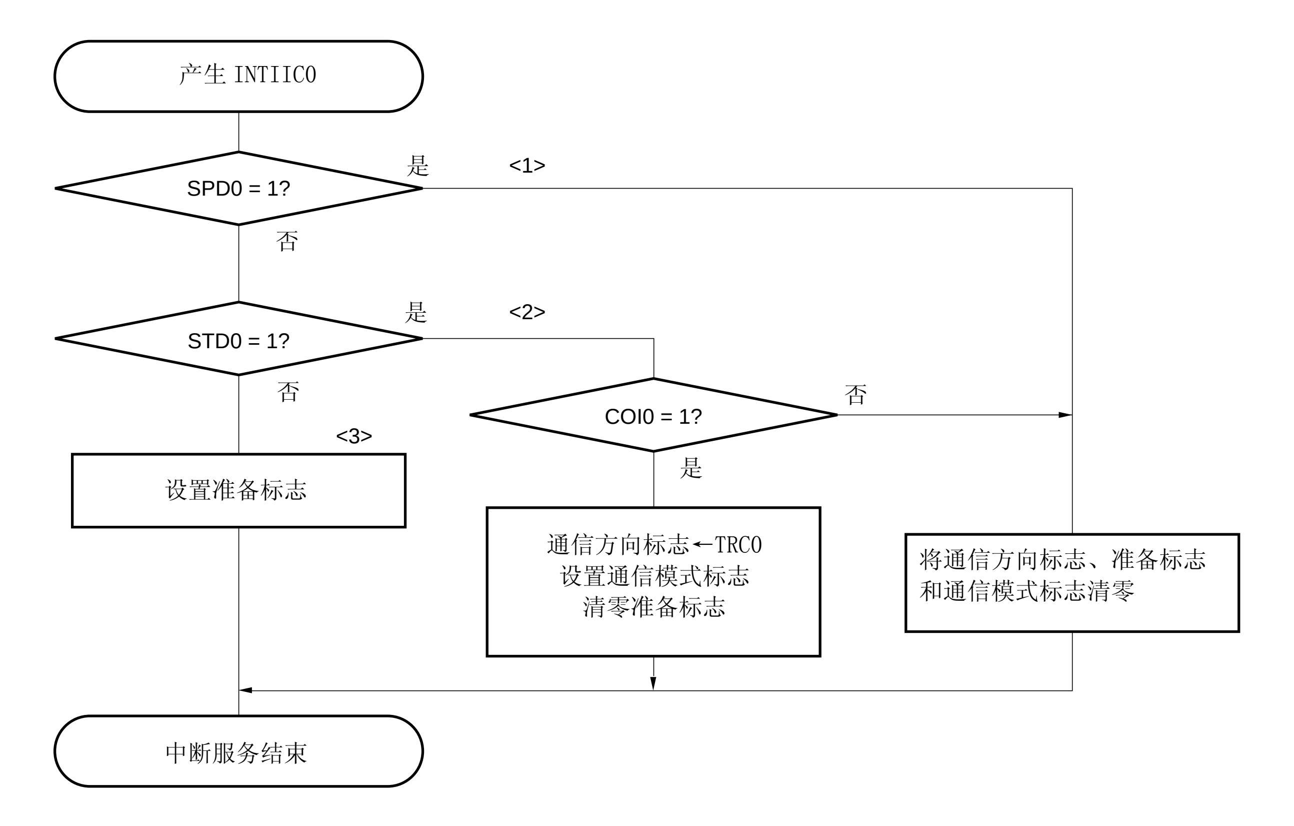

**备注**　遵照正在通信的产品规范，注意发送和接收的格式。

以下示例说明了从设备 INTIIC0 中断处理过程 (处理中假设不使用扩展码)。INTIIC0 中断检查状态，同时执行下列操作。

<1> 若发出停止条件则通信停止。
<2> 若发出起始条件，检查地址并在地址不相等时结束通信。如果地址相等，则设置通信模式，取消等待，且处理过程从中断返回(准备标志清零)。
<3> 对于数据发送/接收，仅需设置准备标志。处理过程从中断返回，同时$I^2C$总线保持等待状态。

**备注**　以上<1> ～ <3> 与**图 16-26 从设备操作流程图(2)**中 <1> ～ <3> 相对应。

★ **图 16-26. 从设备操作流程图(2)**



### 16.5.17 I$^2$C中断请求 (INTIIC0)产生时序

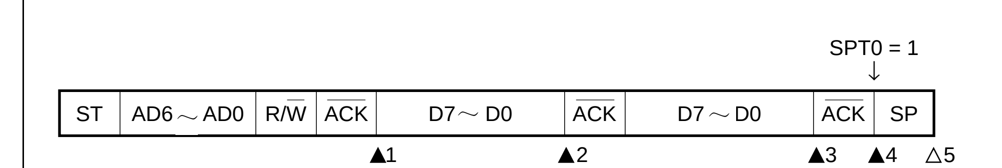
以下显示了发送或接收数据的时序和中断请求信号 INTIIC0 产生时序，以及 INTIIC0 信号产生时 IICS0 寄存器的值。

备注　ST:　起始条件
AD6 ～ AD0:　地址
$R/\overline{W}$:　传送方向指示
$\overline{ACK}$:　应答
D7 ～ D0:　数据
SP:　停止条件

**(1)** 主设备操作

**(a)** 开始～ 地址 ～ 数据 ～ 数据 ～ 停止 (发送/接收)

**(i)** 当 **WTIM0 = 0**

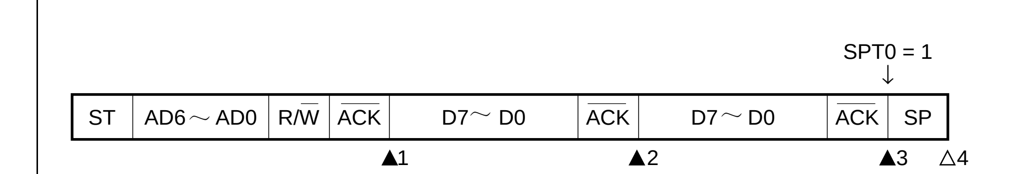



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B
▲3: IICS0 = 1000×000B (设置 WTIM0 为 1)[注]
▲4: IICS0 = 1000××00B (设置SPT0 为 1)[注]
△5: IICS0 = 00000001B

注　设置 WTIM0 为 1 并改变产生 INTIIC0 中断请求信号的时序，可以产生一个停止条件。

备注　▲: 始终产生
△: 仅当 SPIE0 = 1 时产生
×: 不必考虑

**(ii)** 当 **WTIM0 = 1**

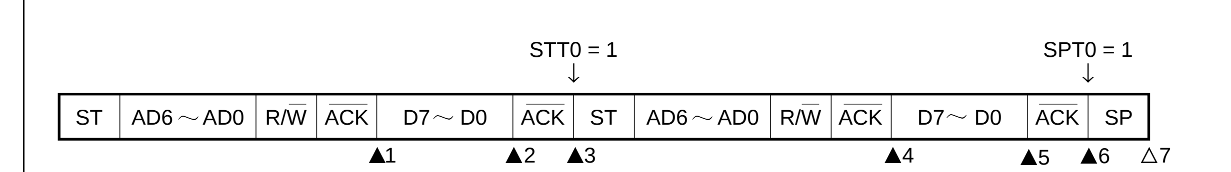



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B
▲3: IICS0 = 1000××00B (设置 SPT0 为 1)
△4: IICS0 = 00000001B

备注　▲: 始终产生
△: 仅当 SPIE0 = 1 时产生
×: 不必考虑

**(b)** 开始 ～ 地址 ～ 数据 ～ 开始 ～ 地址 ～ 数据 ～ 停止 **(**重启**)**

**(i)** 当 **WTIM0 = 0**

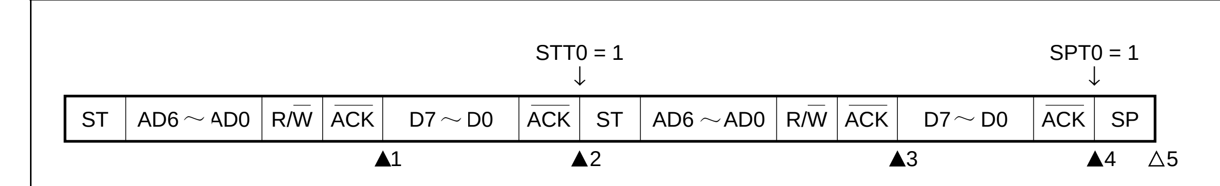



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (设置WTIM0 为 1)[注 1]
▲3: IICS0 = 1000××00B (将 WTIM0 清零[注 2], 设置STT0 为 1)
▲4: IICS0 = 1000×110B
▲5: IICS0 = 1000×000B (设置WTIM0 为 1)[注 3]
▲6: IICS0 = 1000××00B (设置 SPT0 为 1)
△7: IICS0 = 00000001B

注　**1.** 设置 WTIM0 为 1 并改变产生 INTIIC0 中断请求信号的时序，可以产生一个起始条件。
　　**2.** 设置 WTIM0=0 保存原始设置。
　　**3.** 设置 WTIM0 为 1 并改变产生 INTIIC0 中断请求信号的时序，可以产生一个停止条件。

备注　▲: 始终产生
　　　△: 仅当 SPIE0 = 1 时产生
　　　×: 不必考虑

**(ii)** 当 **WTIM0 = 1**

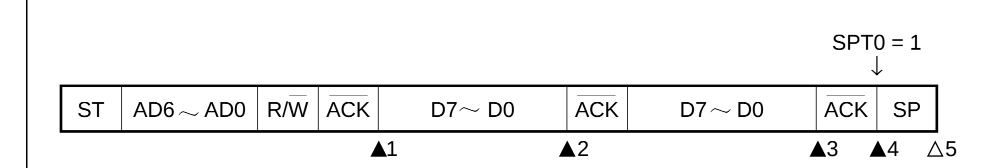



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (设置 STT0 为 1)
▲3: IICS0 = 1000×110B
▲4: IICS0 = 1000××00B (设置 SPT0 为 1)
△5: IICS0 = 00000001B

备注　▲: 始终产生
　　　△: 仅当 SPIE0 = 1 时产生
　　　×: 不必考虑

**(c)** 开始 ～ 代码 ～ 数据 ～ 数据 ～ 停止 **(**发送扩展码**)**

**(i)** 当 **WTIM0 = 0**

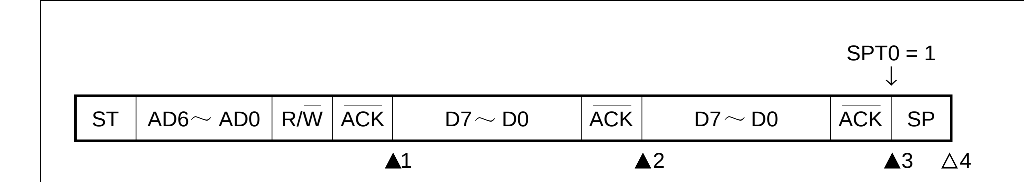



▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×000B
▲3: IICS0 = 1010×000B (设置WTIM0 为 1)[注]
▲4: IICS0 = 1010××00B (设置 SPT0 为 1)
△5: IICS0 = 00000001B

注　设置 WTIM0 为 1 并改变 INTIIC0 产生中断请求信号的时序，可以产生一个停止条件。

备注　▲: 始终产生
△: 仅当 SPIE0 = 1 时产生
×: 不必考虑

**(ii)** 当 **WTIM0 = 1**



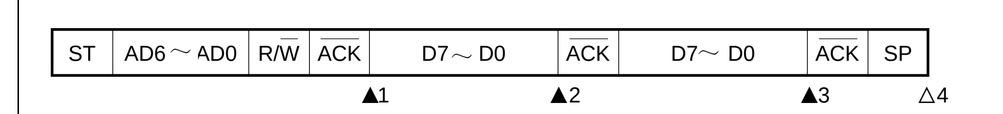

▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×100B
▲3: IICS0 = 1010××00B (设置 SPT0 为 1)
△4: IICS0 = 00001001B

备注　▲: 始终产生
△: 仅当 SPIE0 = 1 时产生
×: 不必考虑

**(2)** 从设备操作 **(**接收从设备地址数据**)**

**(a)** 开始 ～ 地址 ～ 数据 ～ 数据 ～ 停止

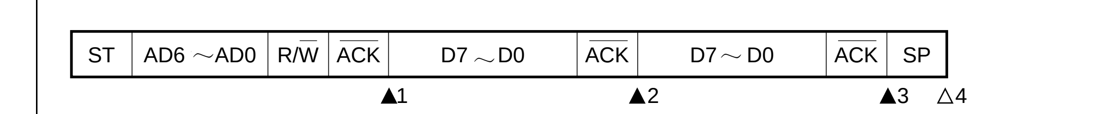
**(i)** 当 **WTIM0 = 0**




▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

备注　▲:　始终产生
△:　仅当 SPIE0 = 1 时产生
×:　不必考虑

**(ii)** 当 **WTIM0 = 1**

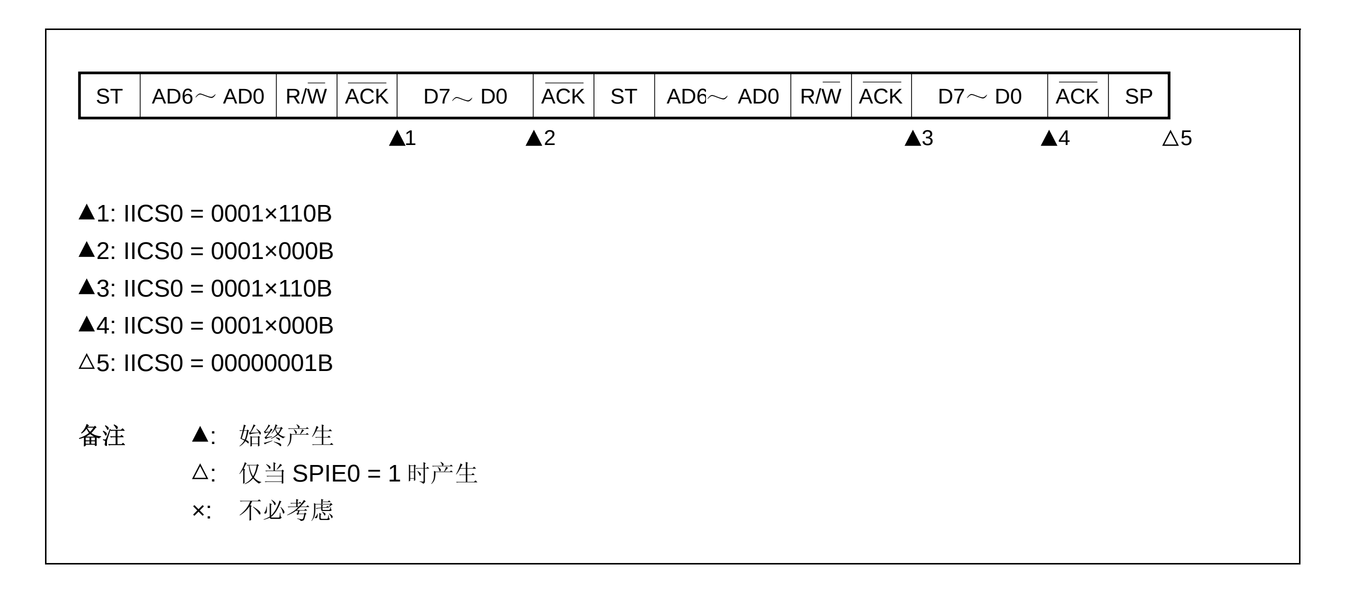



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

备注　▲:　始终产生
△:　仅当 SPIE0 = 1 时产生
×:　不必考虑

**(b)** 开始 ～ 地址 ～ 数据 ～ 开始 ～ 地址 ～ 数据 ～ 停止

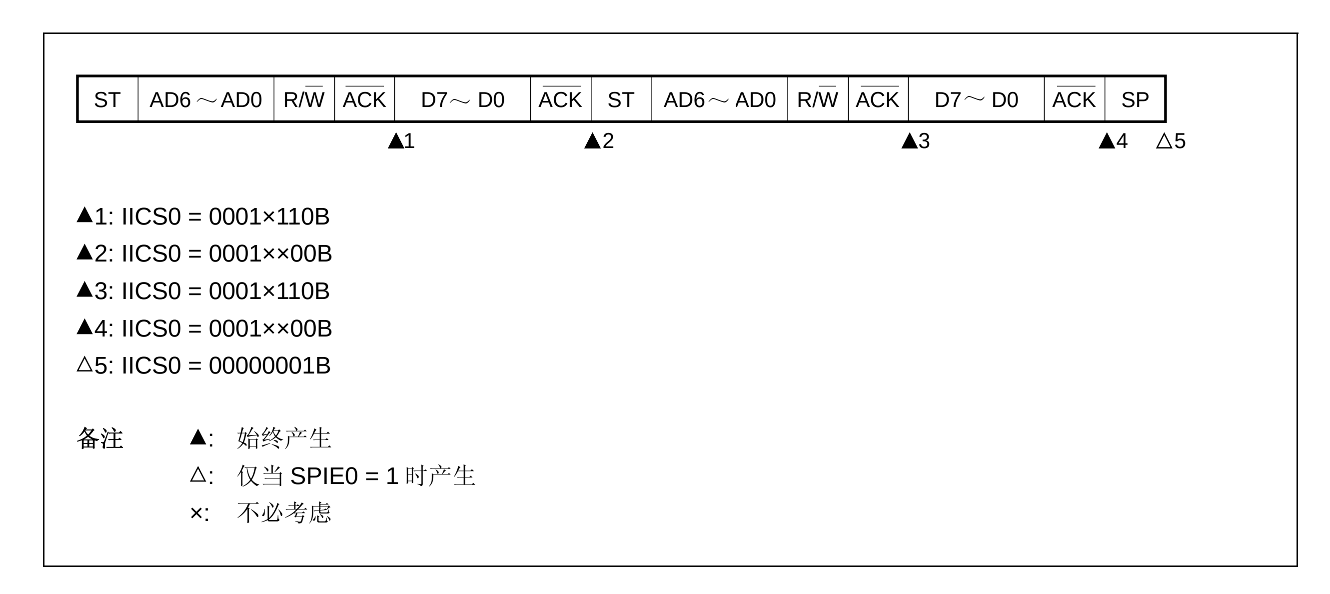
**(i)** 当 **WTIM0 = 0** (重启后，与 **SVA0** 相等)




▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001×000B
△5: IICS0 = 00000001B

备注 ▲: 始终产生
△: 仅当 SPIE0 = 1 时产生
×: 不必考虑

**(ii)** 当 **WTIM0 = 1** (重启后，与 **SVA0** 相等)

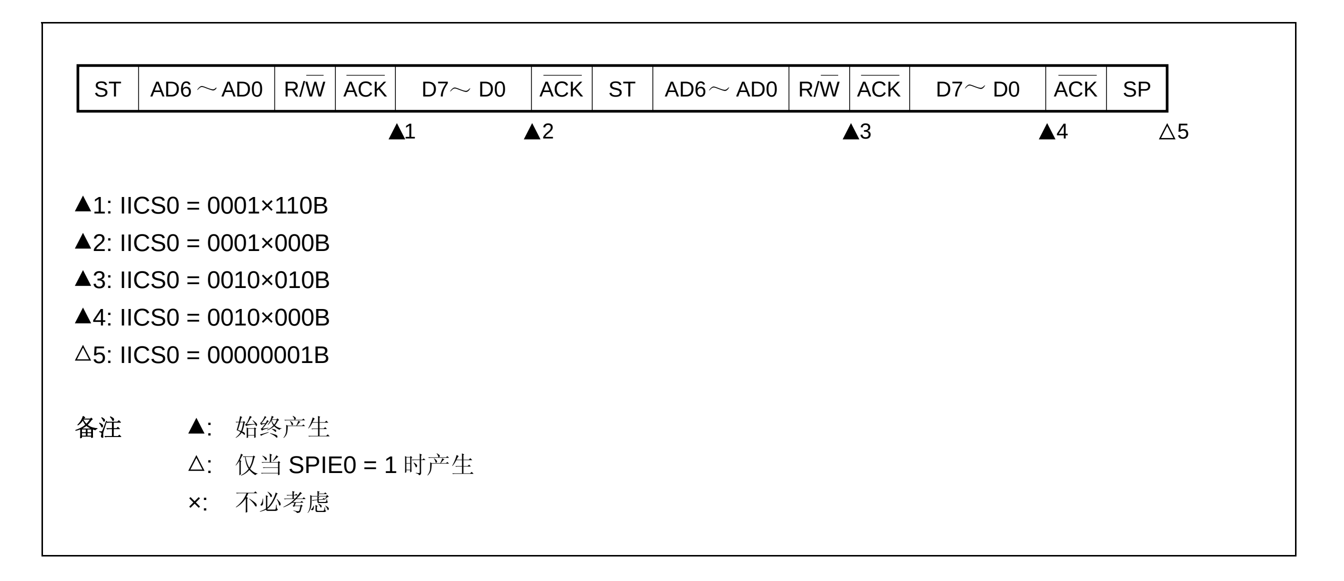



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001××00B
△5: IICS0 = 00000001B

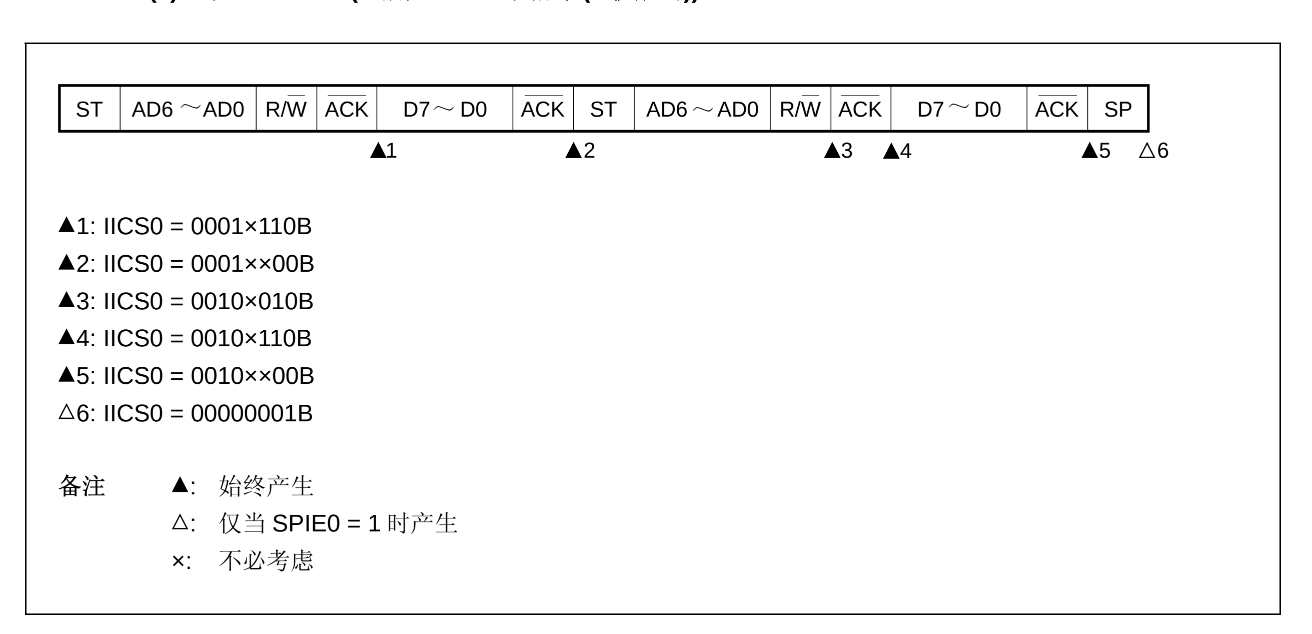
备注 ▲: 始终产生
△: 仅当 SPIE0 = 1 时产生
×: 不必考虑

**(c)** 开始 ～ 地址 ～ 数据 ～ 开始 ～ 代码 ～ 数据 ～ 停止

**(i)** 当 **WTIM0 = 0 (**重启后，地址不相等 **(=** 扩展码**))**

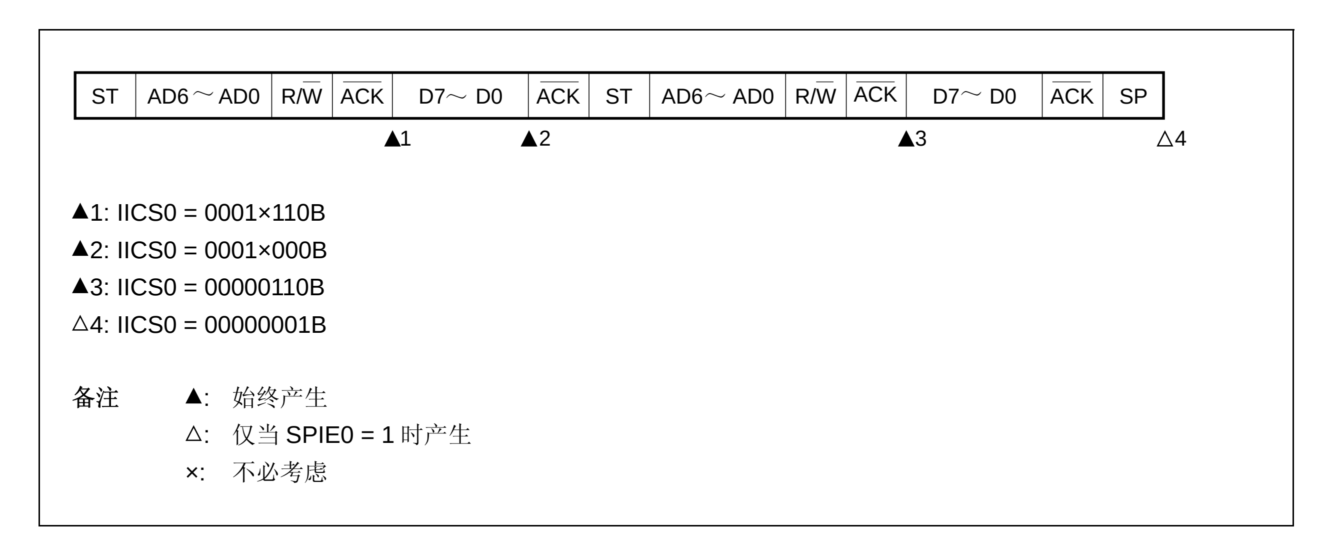



**(ii)** 当 **WTIM0 = 1 (**重启后，地址不相等 **(=** 扩展码**))**



**(d)** 开始 ～ 地址 ～ 数据 ～ 开始 ～ 地址 ～ 数据 ～ 停止

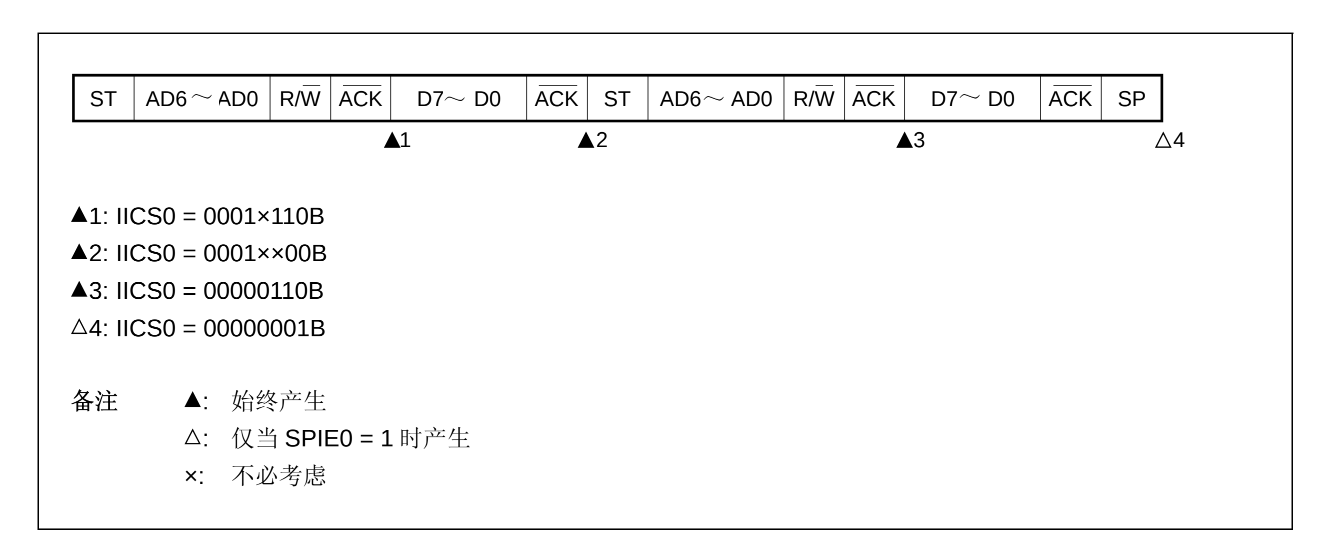
**(i)** 当 **WTIM0 = 0 (**重启后，地址不相等 **(=** 非扩展码**))**




▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 00000110B
△4: IICS0 = 00000001B

备注　▲: 始终产生
△: 仅当 SPIE0 = 1 时产生
×: 不必考虑

**(ii)** 当 **WTIM0 = 1 (**重启后，地址不相等 **(=** 非扩展码**))**




▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 00000110B
△4: IICS0 = 00000001B

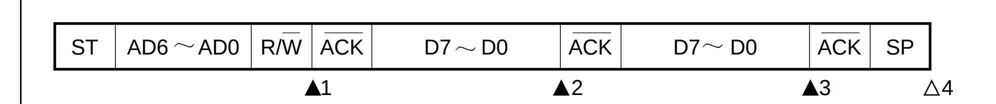
备注　▲: 始终产生
△: 仅当 SPIE0 = 1 时产生
×: 不必考虑

**(3)** 从设备操作 (当接收到扩展码时)

接收到扩展码时，设备总是参与通信。

**(a)** 开始 ～ 代码 ～ 数据 ～ 数据 ～ 停止

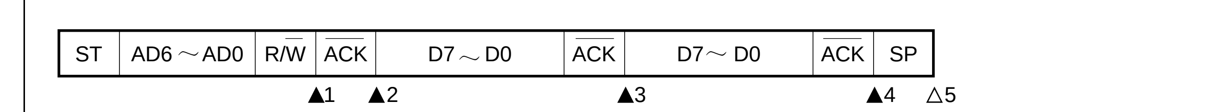
**(i)** 当 **WTIM0 = 0**




▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

备注　▲: 始终产生
　　　△: 仅当 SPIE0 = 1 时产生
　　　×: 不必考虑

**(ii)** 当 **WTIM0 = 1**

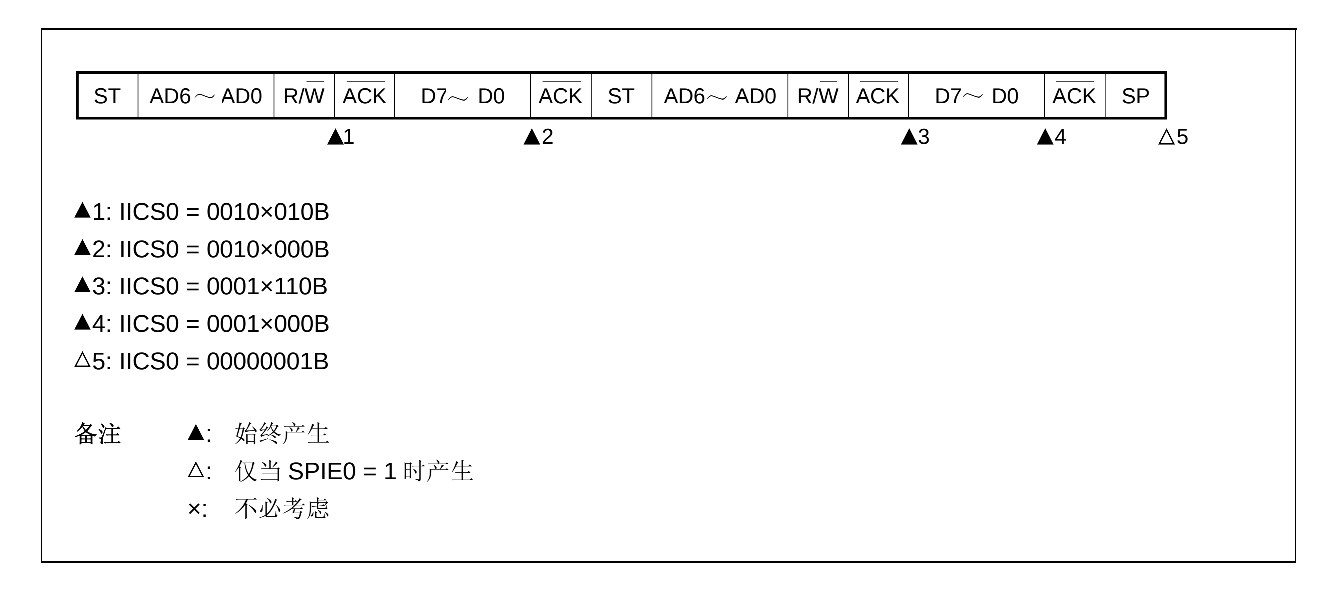



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

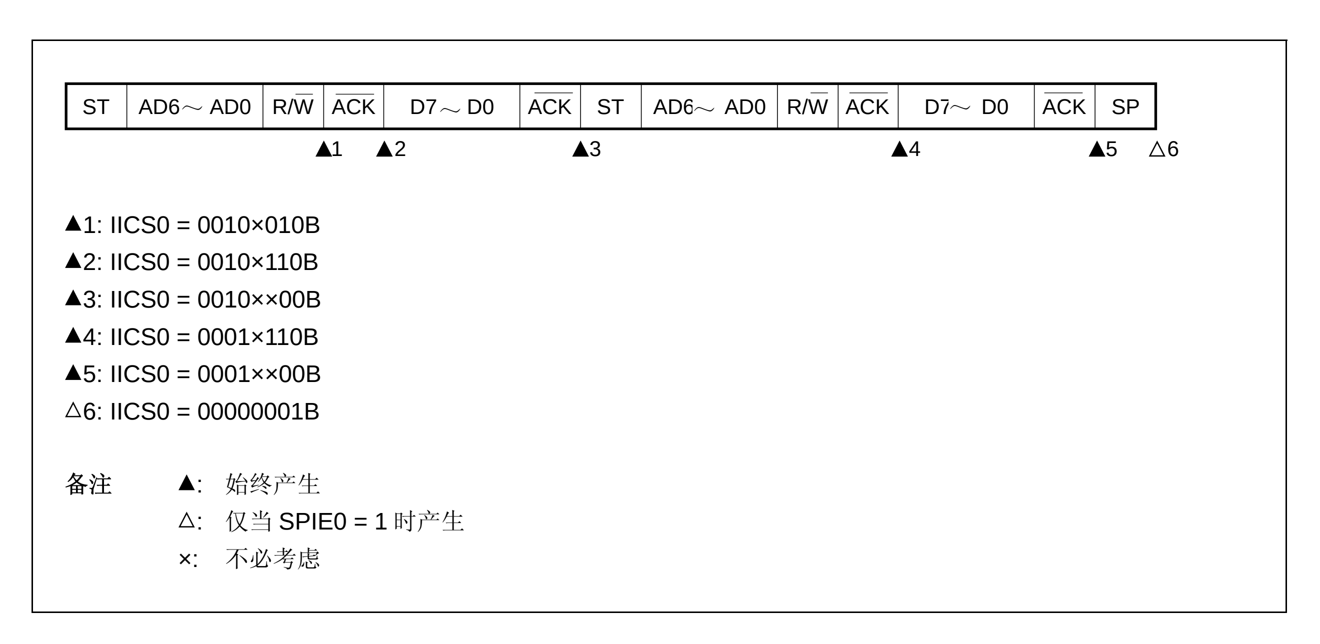
备注　▲: 始终产生
　　　△: 仅当 SPIE0 = 1 时产生
　　　×: 不必考虑

**(b)** 开始 ～ 代码 ～ 数据 ～ 开始 ～ 地址 ～ 数据 ～ 停止

**(i)** 当 **WTIM0 = 0 (**重启后，与 **SVA0** 相等**)**

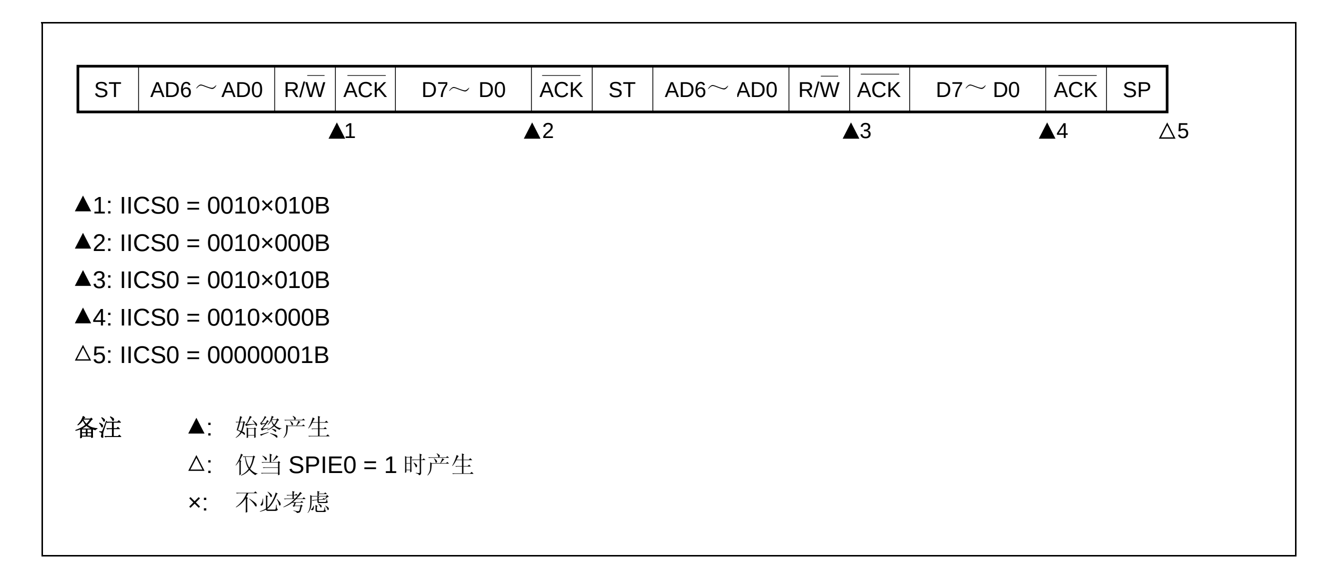



**(ii)** 当 **WTIM0 = 1 (**重启后，与 **SVA0** 相等**)**

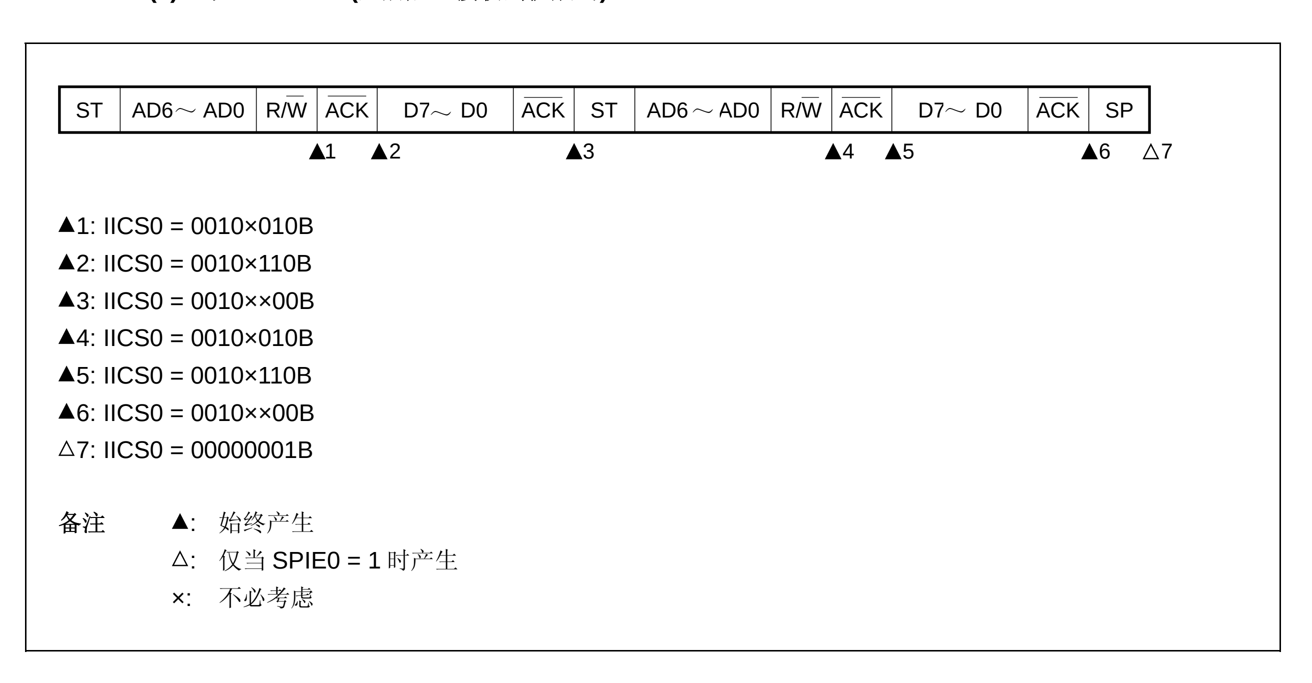


**(c)** 开始 ～ 代码 ～ 数据 ～ 开始 ～ 代码 ～ 数据 ～ 停止

**(i)** 当 **WTIM0 = 0 (**重启后，接收到扩展码**)**

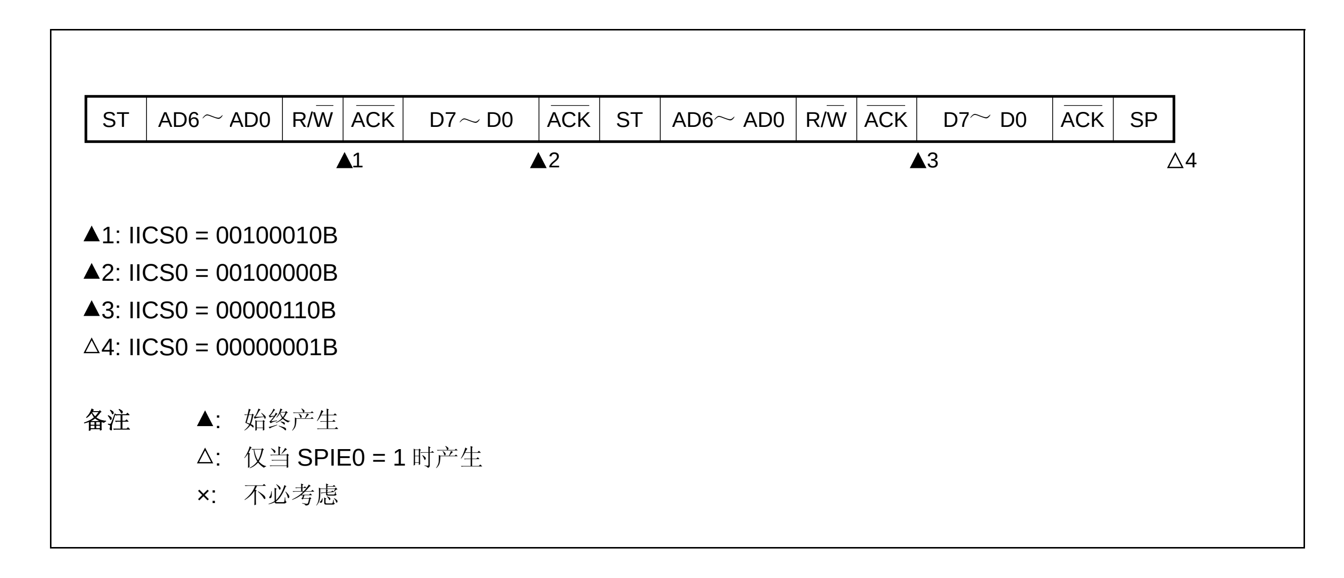



**(ii)** 当 **WTIM0 = 1 (**重启后，接收到扩展码**)**

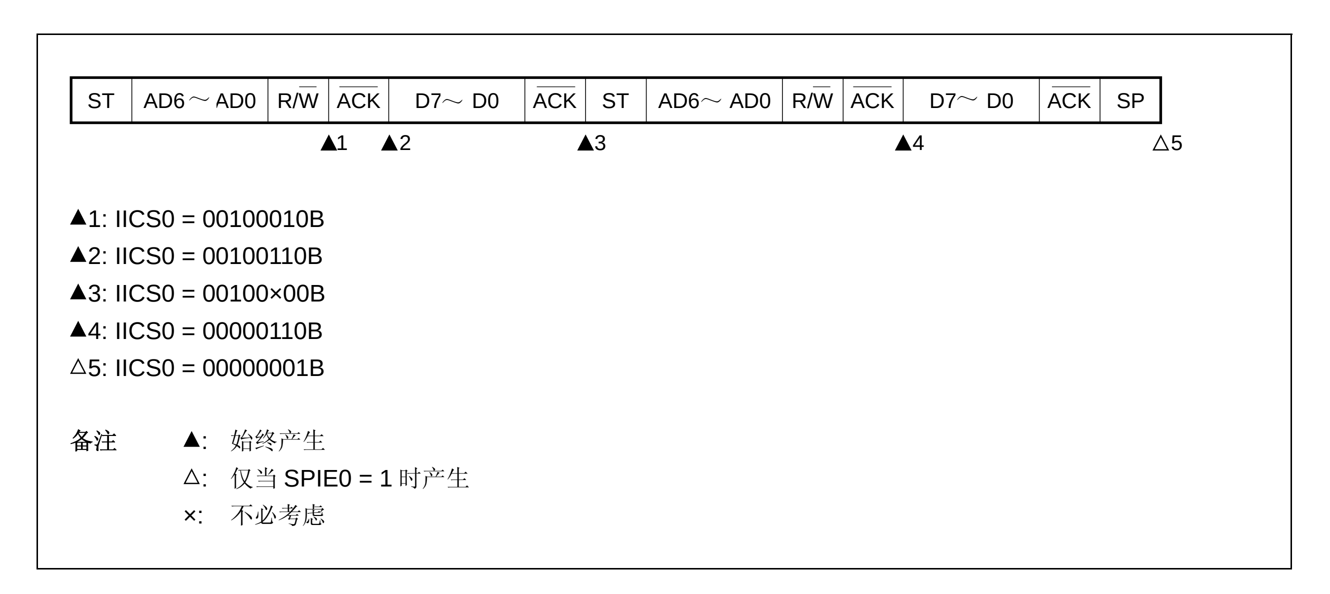


**(d)** 开始 ～ 代码 ～ 数据 ～ 开始 ～ 地址 ～ 数据 ～ 停止

**(i)** 当 **WTIM0 = 0 (**重启后，地址不相等 **(=** 非扩展码**))**




**(ii)** 当 **WTIM0 = 1 (**重启后，地址不相等 **(=** 非扩展码**))**



**(4)** 未进行通信时的操作

**(a)** 开始 ～ 代码 ～ 数据 ～ 数据 ～ 停止

| ST | AD6～AD0 | R/$\overline{W}$ | $\overline{ACK}$ | D7～D0 | $\overline{ACK}$ | D7～D0 | $\overline{ACK}$ | SP |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | △1 |

△1: IICS0 = 00000001B

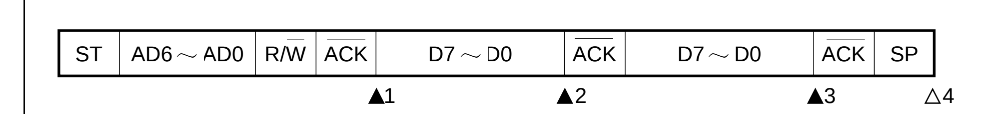
备注　△:　仅当 SPIE0 = 1 时产生

**(5)** 发生仲裁失败时的操作 **(**仲裁失败后作为从设备操作**)**

在多主设备系统中当一个设备作为主设备使用时，可在每次中断请求信号 INTIIC0 产生时读取 MSTS0 位来检查仲裁结果。

**(a)** 发送从设备地址数据期间发生仲裁失败时

**(i)**　当 **WTIM0 = 0**




▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

备注　▲:　始终产生
　　　△:　仅当 SPIE0 = 1 时产生
　　　×:　不必考虑

(ii) 当 WTIM0 = 1

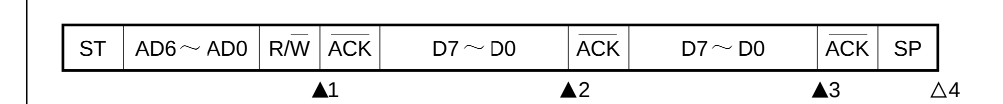
| ST | AD6～AD0 | R/$\overline{W}$ | $\overline{ACK}$ | D7～D0 | $\overline{ACK}$ | D7～D0 | $\overline{ACK}$ | SP |
|---|---|---|---|---|---|---|---|---|
| | | | | ▲1 | ▲2 | | ▲3 | △4 |

▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

备注 ▲: 始终产生
△: 仅当 SPIE0 = 1 时产生
×: 不必考虑

(b) 发送扩展码期间发生仲裁失败时

(i) 当 WTIM0 = 0




▲1: IICS0 = 0110×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

备注 ▲: 始终产生
△: 仅当 SPIE0 = 1 时产生
×: 不必考虑

**(ii)　当 WTIM0 = 1**

| ST | AD6～AD0 | R/$\overline{W}$ | $\overline{ACK}$ | D7～D0 | $\overline{ACK}$ | D7～D0 | $\overline{ACK}$ | SP |
|---|---|---|---|---|---|---|---|---|
| | | ▲1 | ▲2 | | ▲3 | | ▲4 | △5 |

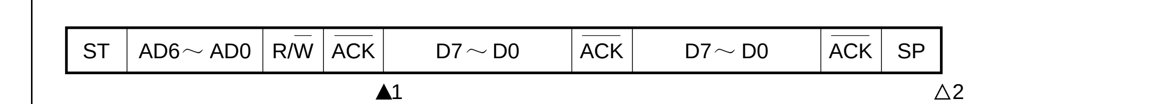
▲1: IICS0 = 0110×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

备注：　▲:　始终产生
　　　　△:　仅当 SPIE0 = 1 时产生
　　　　×:　不必考虑

**(6)　发生仲裁失败时的操作 (仲裁失败后不进行通信)**

在多主设备系统中当一个设备作为主设备使用时，可在每次中断请求信号 INTIIC0 产生时读取 MSTS0 位来检查仲裁结果。

**(a)　发送从设备地址数据期间发生仲裁失败时 (当 WTIM0 = 1)**




▲1: IICS0 = 01000110B
△2: IICS0 = 00000001B

备注：　▲:　始终产生
　　　　△:　仅当 SPIE0 = 1 时产生

**(b)** 发送扩展码期间发生仲裁失败时

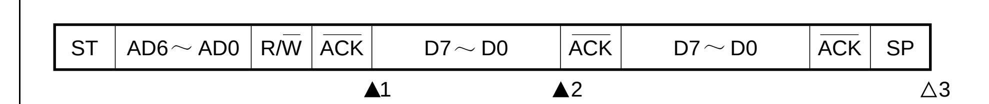
| ST | AD6～AD0 | R/$\overline{W}$ | $\overline{ACK}$ | D7～D0 | $\overline{ACK}$ | D7～D0 | $\overline{ACK}$ | SP |
|---|---|---|---|---|---|---|---|---|
| | | | ▲1 | | | | | △2 |

▲1: IICS0 = 0110×010B
通过软件设置 LREL0 = 1
△2: IICS0 = 00000001B

备注　▲:　始终产生
　　　△:　仅当 SPIE0 = 1 时产生
　　　×:　不必考虑

**(c)** 发送数据期间发生仲裁失败时

**(i)**　当 **WTIM0 = 0**




▲1: IICS0 = 10001110B
▲2: IICS0 = 01000000B
△3: IICS0 = 00000001B

备注　▲:　始终产生
　　　△:　仅当 SPIE0 = 1 时产生

**(ii) 当 WTIM0 = 1**

| ST | AD6～AD0 | R/$\overline{W}$ | $\overline{ACK}$ | D7～D0 | $\overline{ACK}$ | D7～D0 | $\overline{ACK}$ | SP |
|---|---|---|---|---|---|---|---|---|
| | | | ▲1 | | ▲2 | | | △3 |

▲1: IICS0 = 10001110B
▲2: IICS0 = 01000100B
△3: IICS0 = 00000001B

备注 ▲: 始终产生
△: 仅当 SPIE0 = 1 时产生

**(d) 数据传输期间由于重启条件导致仲裁失败时**

**(i) 非扩展码 (例: 与 SVA0 不相等)**




▲1: IICS0 = 1000×110B
▲2: IICS0 = 01000110B
△3: IICS0 = 00000001B

备注 ▲: 始终产生
△: 仅当 SPIE0 = 1 时产生
×: 不必考虑
n = 6 ～ 0

**(ii)** 扩展码

| ST | AD6～AD0 | R/$\overline{W}$ | $\overline{ACK}$ | D7～Dn | ST | AD6～AD0 | R/$\overline{W}$ | $\overline{ACK}$ | D7～D0 | $\overline{ACK}$ | SP |
|---|---|---|---|---|---|---|---|---|---|---|---|

▲1　▲2　△3

▲1: IICS0 = 1000×110B
▲2: IICS0 = 01100010B
通过软件设置 LREL0 = 1
△3: IICS0 = 00000001B

备注　▲:　始终产生
△:　仅当 SPIE0 = 1 时产生
×:　不必考虑
n = 6 ～ 0

**(e)** 数据传送期间由于停止条件导致仲裁失败时

| ST | AD6～AD0 | R/$\overline{W}$ | $\overline{ACK}$ | D7～Dn | SP |
|---|---|---|---|---|---|

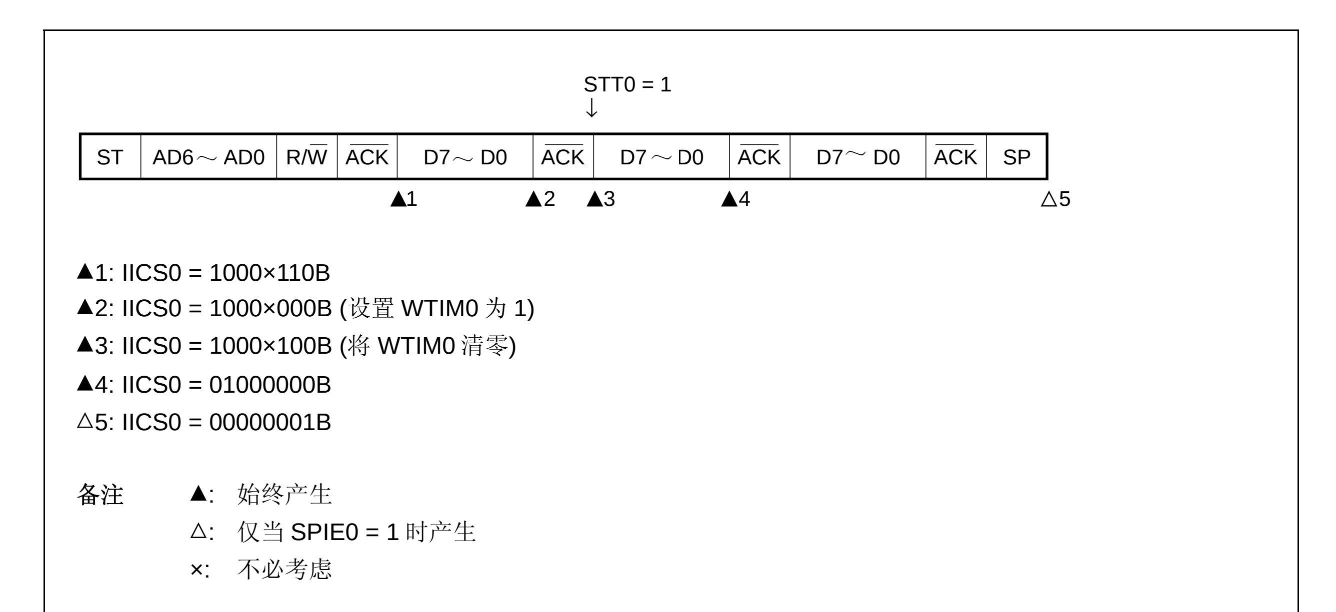
▲1　△2

▲1: IICS0 = 10000110B
△2: IICS0 = 01000001B

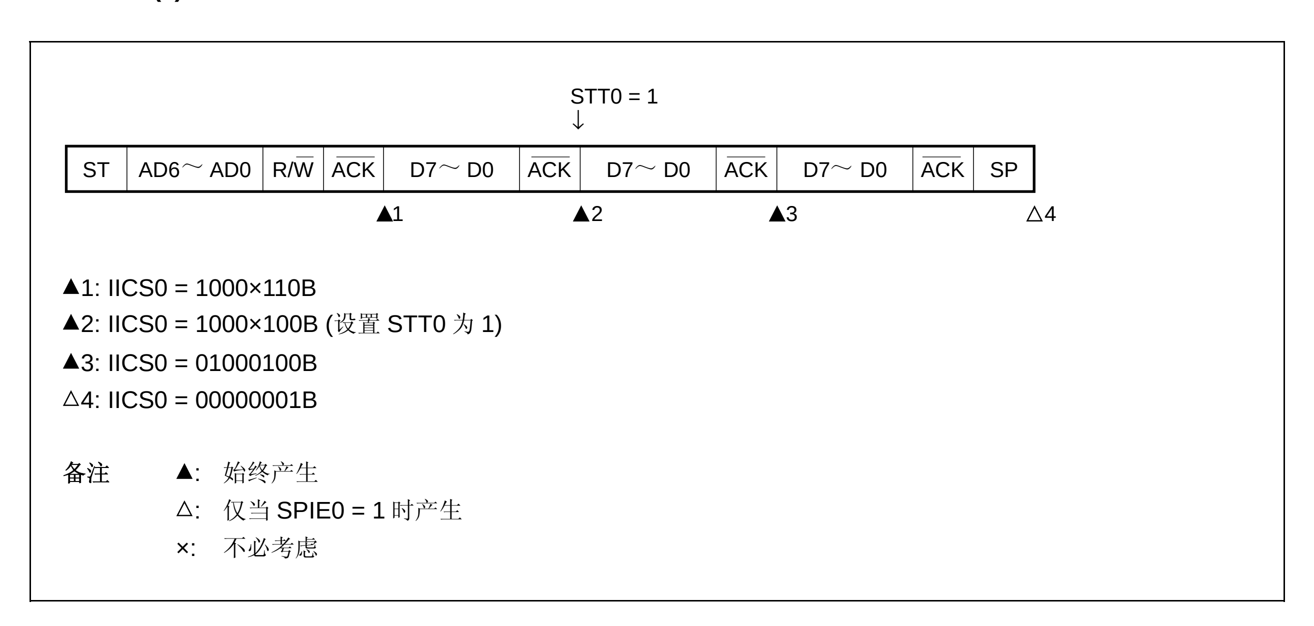
备注　▲:　始终产生
△:　仅当 SPIE0 = 1 时产生
×:　不必考虑
n = 6 ～ 0

**(f)** 要产生重启条件时，由于低电平数据导致仲裁失败时

**(i)** 当 **WTIM0 = 0**

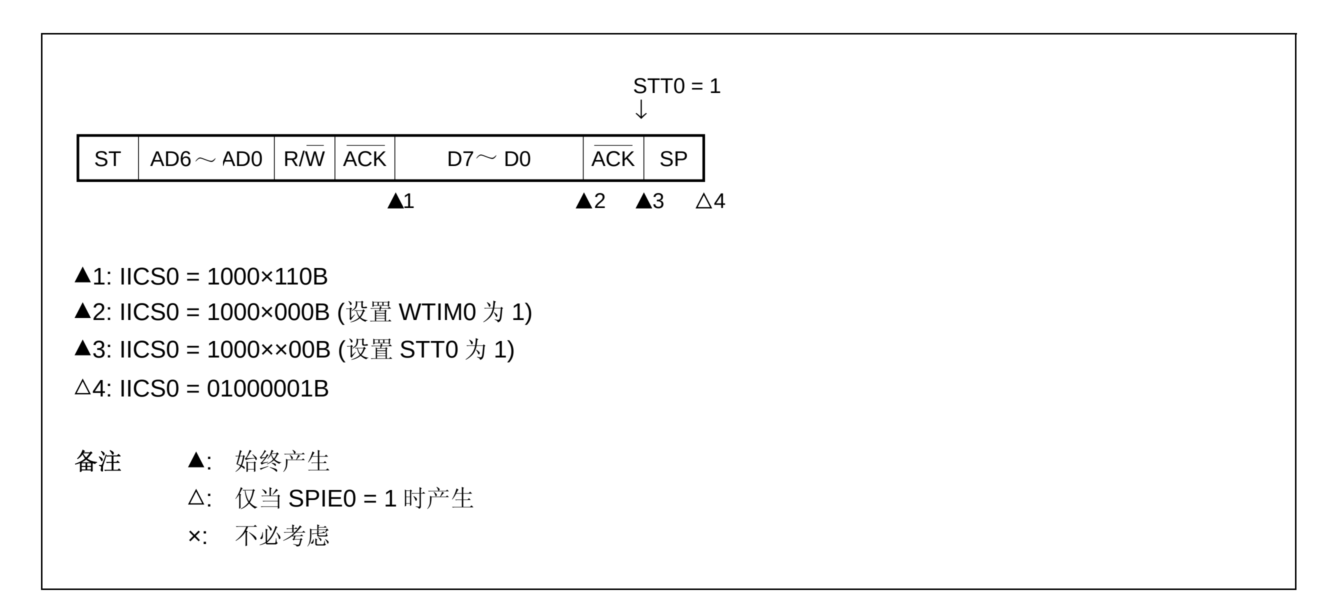



**(ii)** 当 **WTIM0 = 1**



**(g)** 要产生重启条件时，由于停止条件导致仲裁失败时

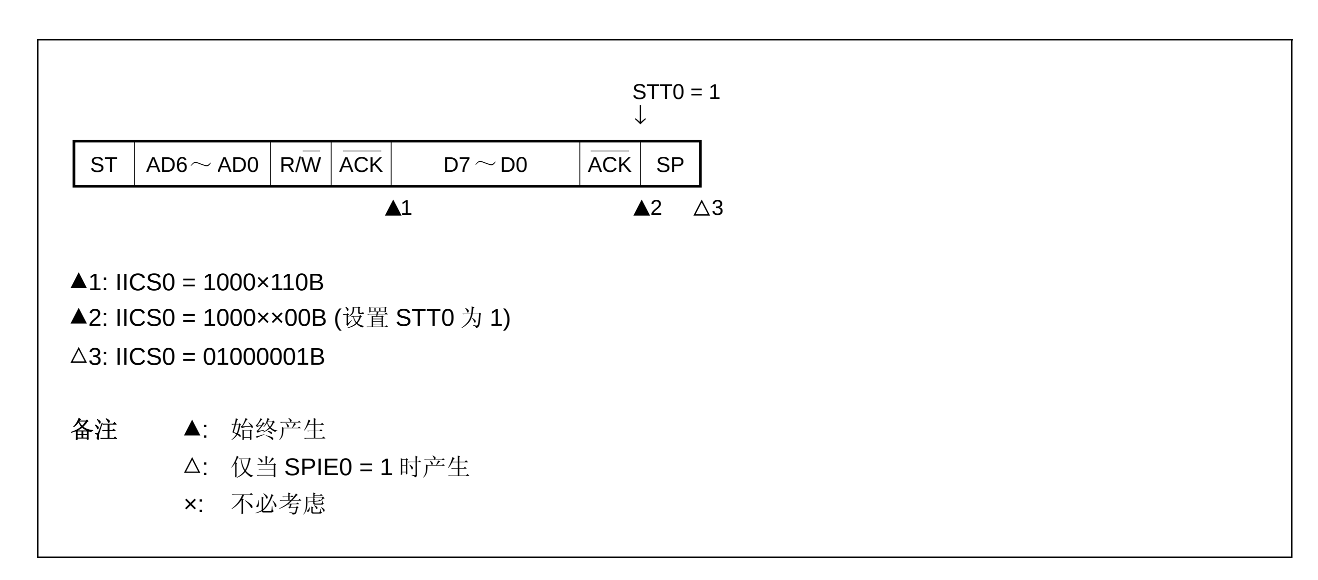
**(i)** 当 **WTIM0 = 0**




▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (设置 WTIM0 为 1)
▲3: IICS0 = 1000××00B (设置 STT0 为 1)
△4: IICS0 = 01000001B

备注　▲: 始终产生
　　　△: 仅当 SPIE0 = 1 时产生
　　　×: 不必考虑

**(ii)** 当 **WTIM0 = 1**

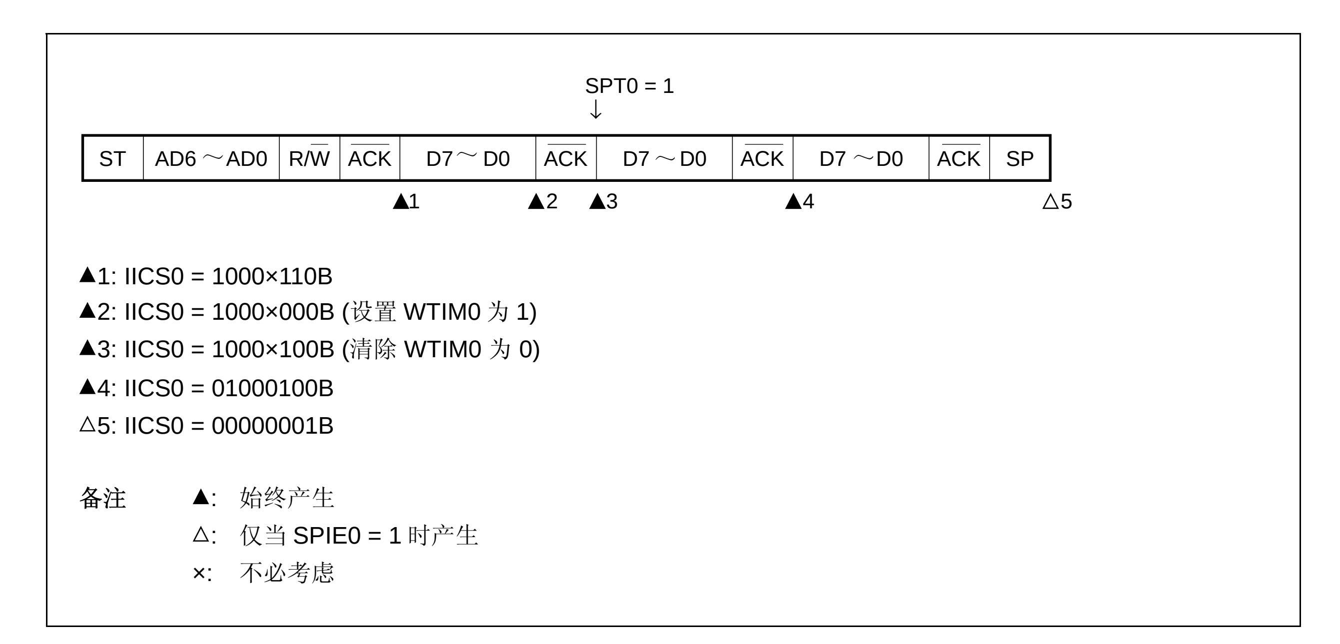



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (设置 STT0 为 1)
△3: IICS0 = 01000001B

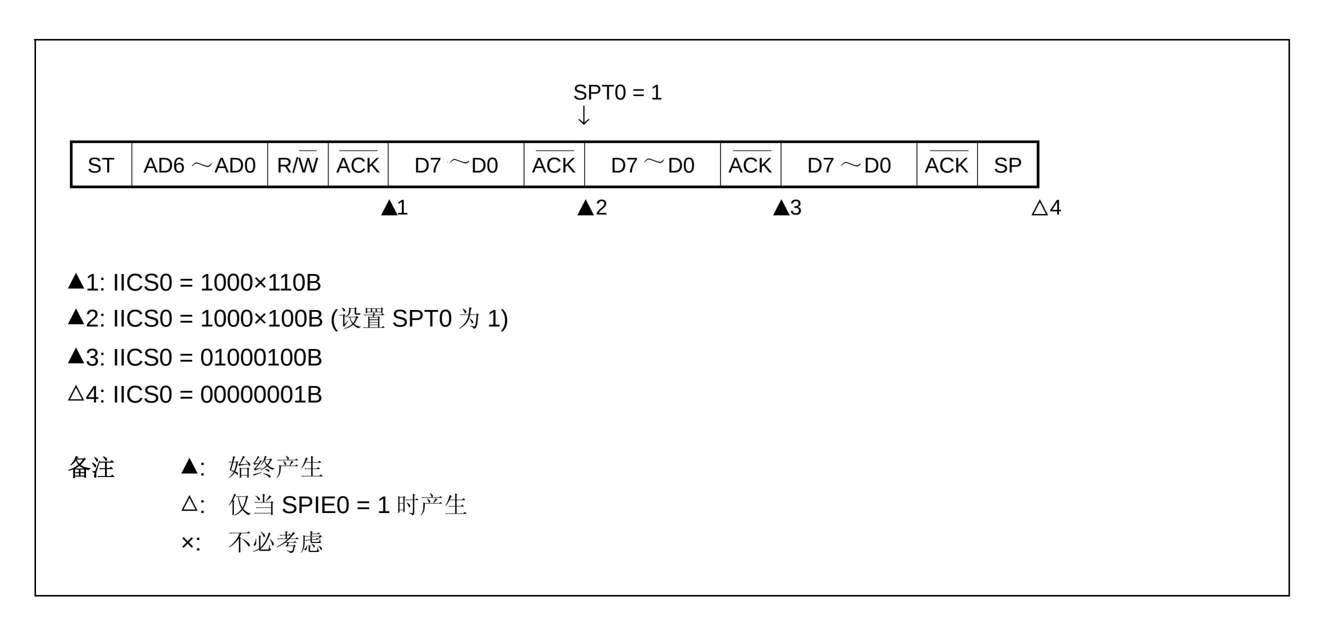
备注　▲: 始终产生
　　　△: 仅当 SPIE0 = 1 时产生
　　　×: 不必考虑

**(h)** 要产生停止条件时，由于低电平数据导致仲裁失败时

**(i)** 当 **WTIM0 = 0**




**(ii)** 当 **WTIM0 = 1**



## 16.6 时序图

使用$I^2C$总线模式时，主设备通过串行总线输出一个地址来选择一个从设备作为它的通信伙伴。

输出从设备地址后，主设备发送 TRC0 位(IIC 状态寄存器 0 (IICS0)的第 3 位)，用于指定数据传送方向，然后开始与从设备进行串行通信。

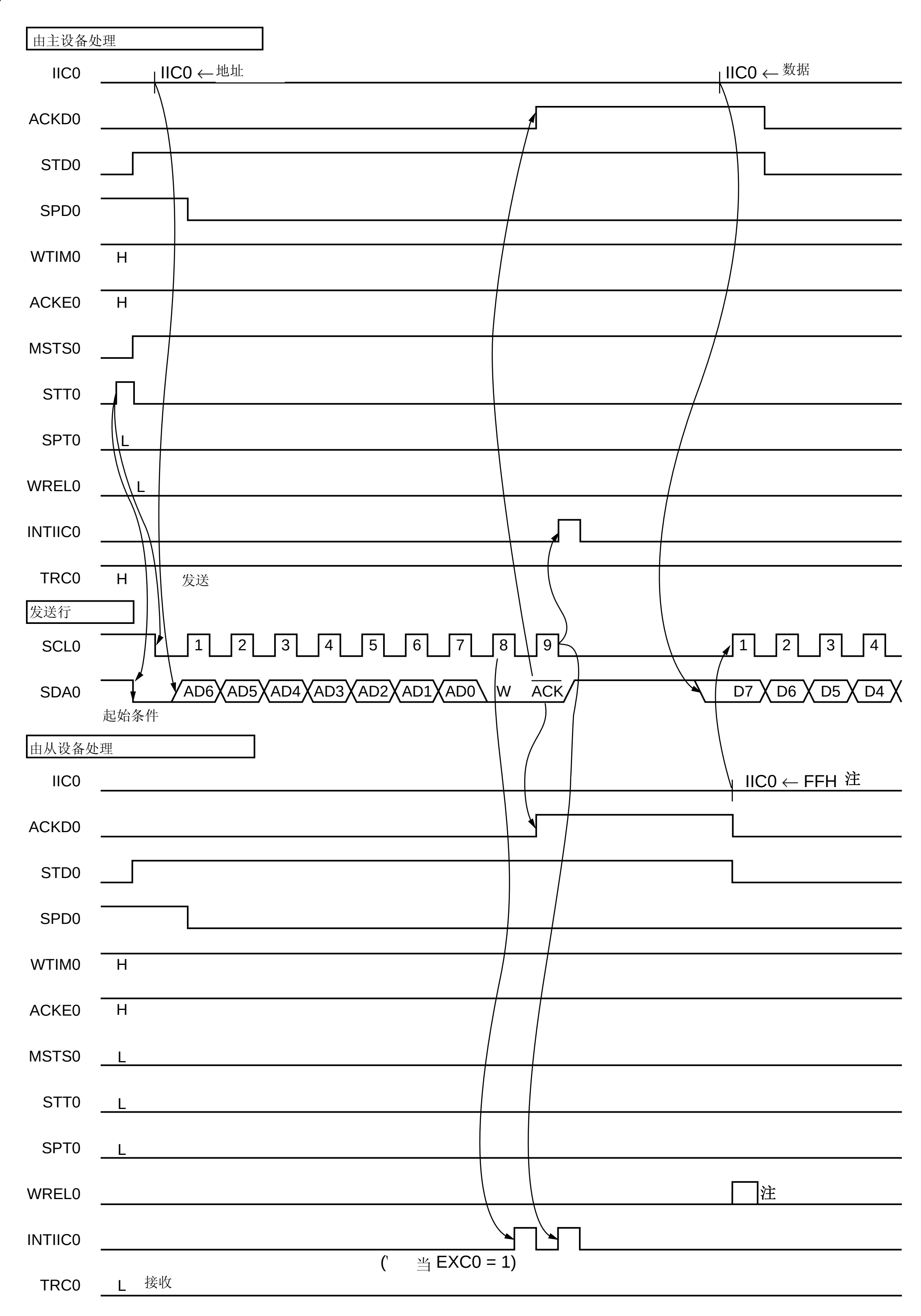
图 16-27 和 16-28 表示了数据通信的时序图。

IIC 移位寄存器 0 (IIC0)的移位操作和串行时钟(SCL0)的下降沿同步。将发送数据传入 SO0 锁存器并通过 SDA0 引脚输出(MSBfirst)。

在 SCL0 的上升沿通过 SDA0 引脚输入的数据被捕捉到 IIC0 寄存器。

**图 16-27. 主设备至从设备的通信举例**
**(当主设备和从设备都选择 9-时钟等待时) (1/3)**

**(1)** 起始条件 ～ 地址

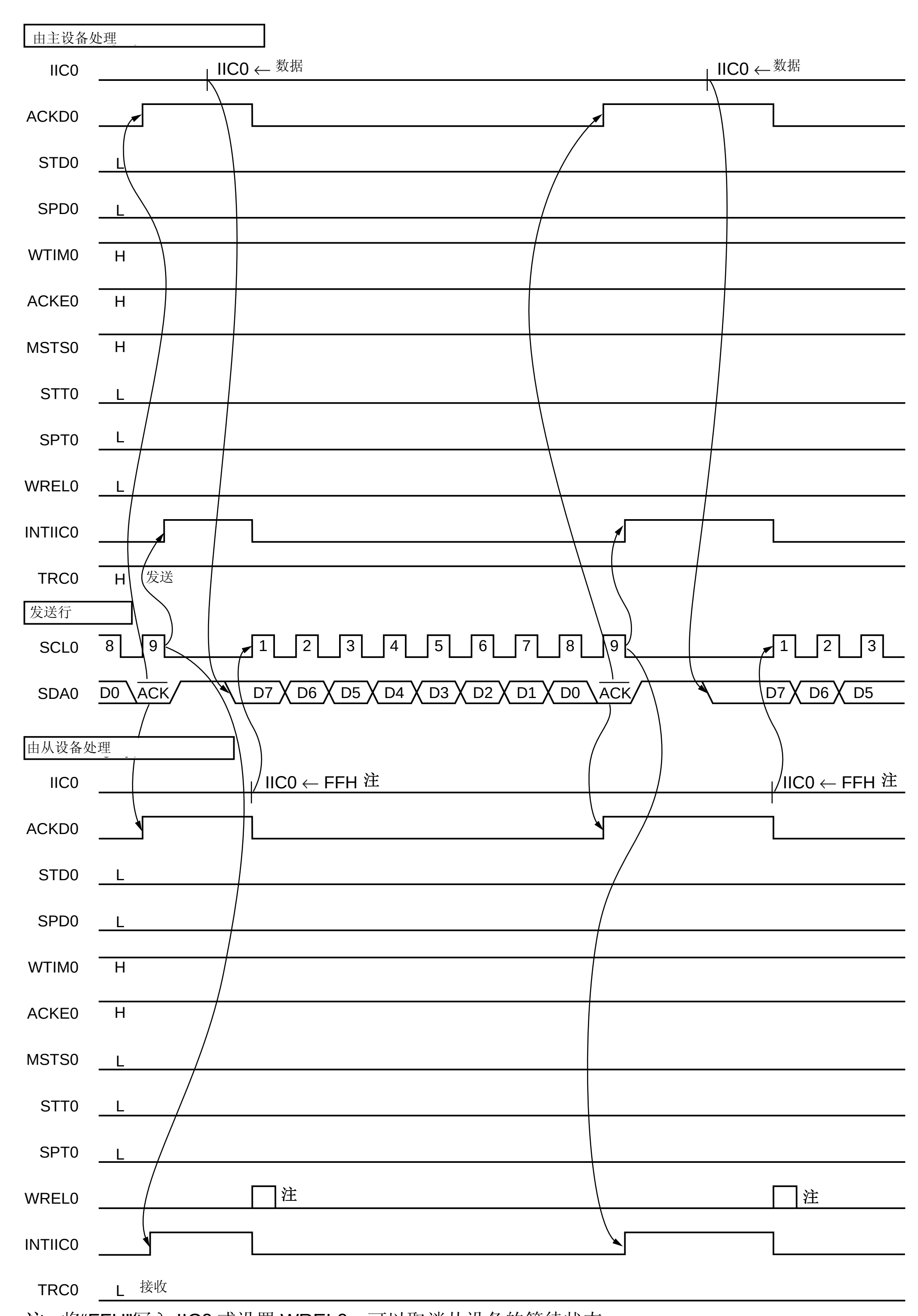



注　　将“FFH”写入 IIC0 或设置 WREL0，可以取消从设备的等待状态。

图 **16-27.** 主设备至从设备的通信举例

**(**当主设备和从设备都选择 **9-**时钟等待时**) (2/3)**

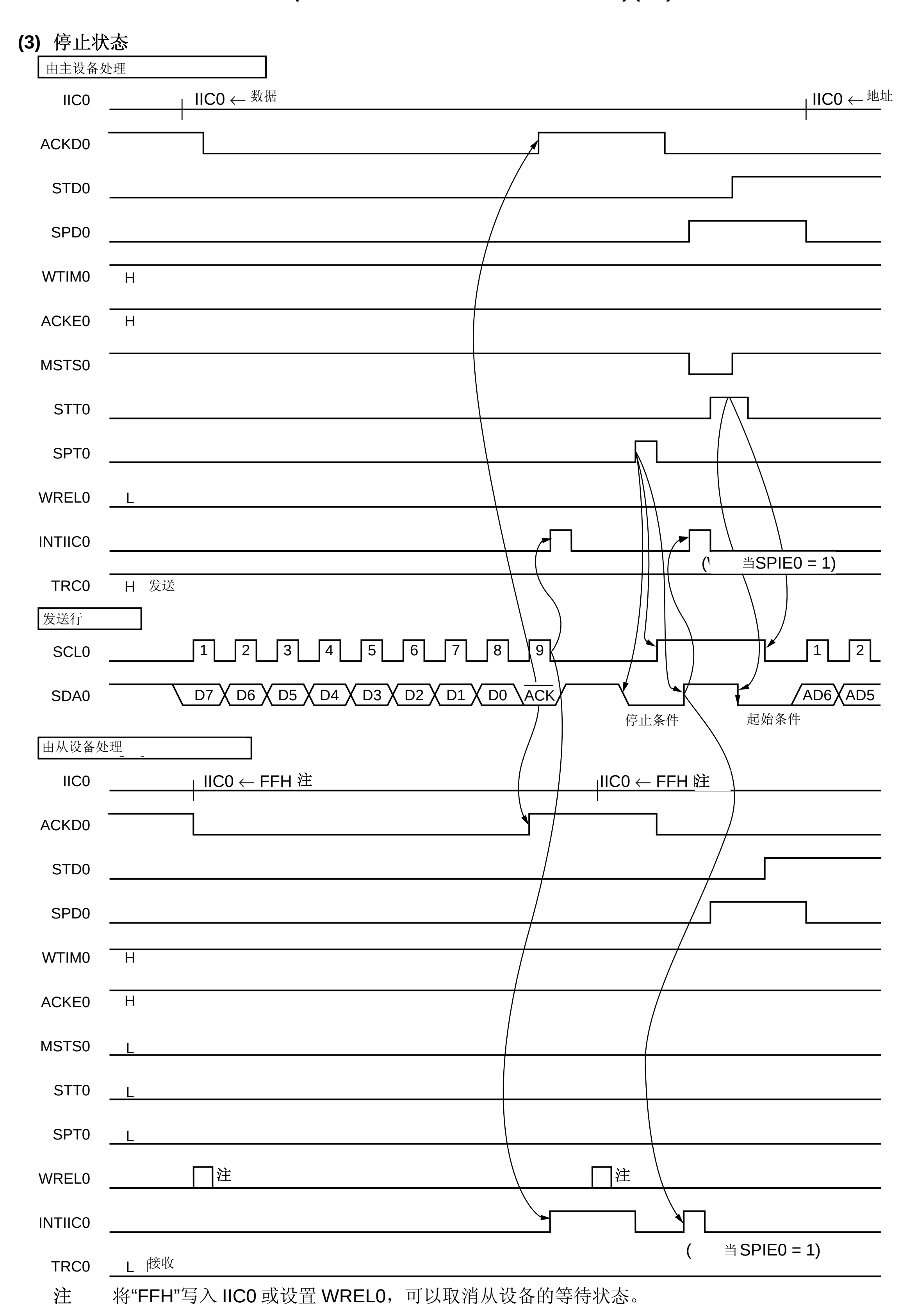
**(2)** 数据




注　将“FFH”写入 IIC0 或设置 WREL0，可以取消从设备的等待状态。

**图 16-27. 主设备至从设备的通信举例**
**(当主设备和从设备都选择 9-时钟等待时) (3/3)**

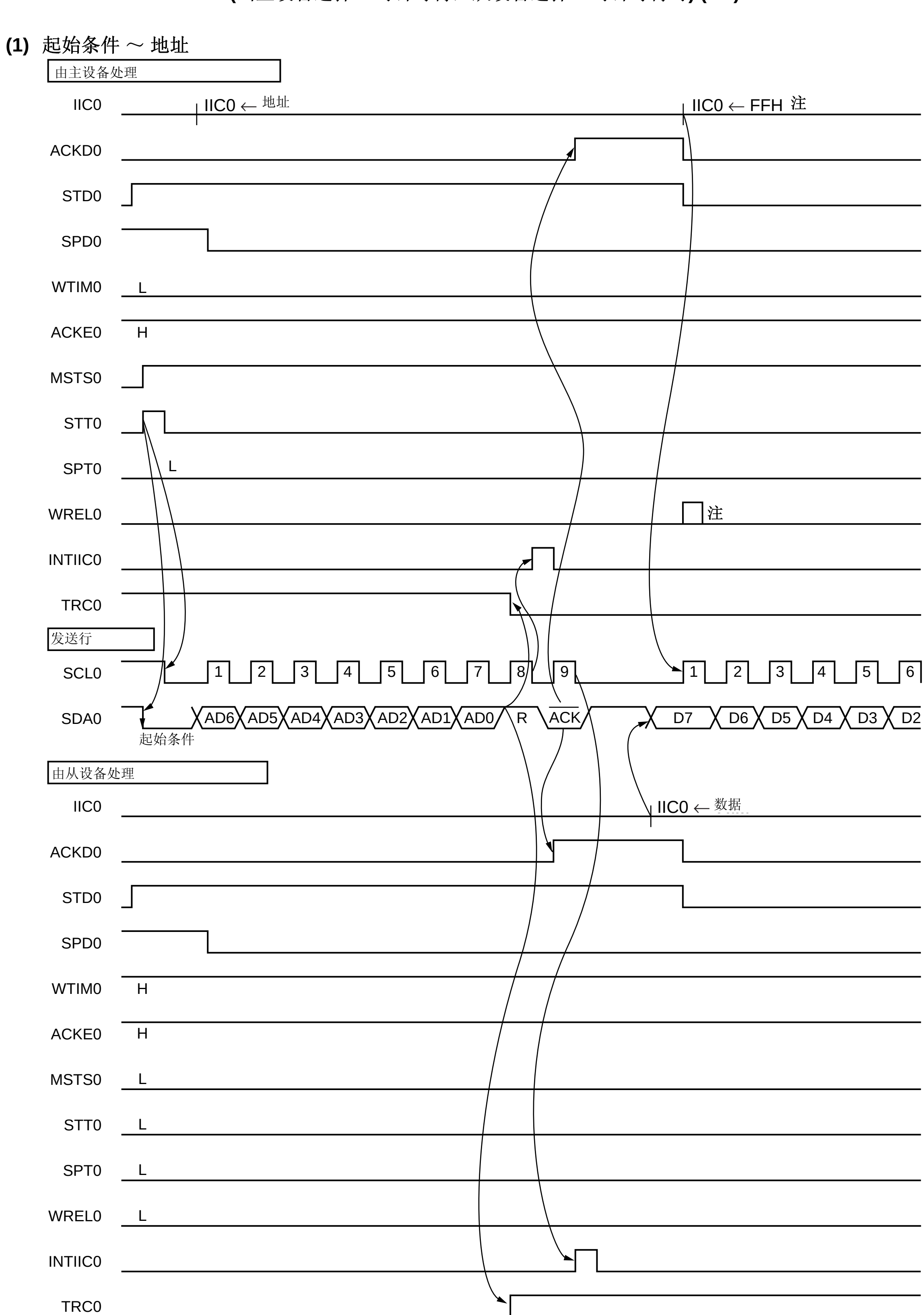
**(3) 停止状态**




注 将“FFH”写入 IIC0 或设置 WREL0，可以取消从设备的等待状态。

图 **16-28.** 从设备至主设备通信举例
**(**当主设备选择 **8-**时钟等待，从设备选择 **9-**时钟等待时**) (1/3)**

**(1)** 起始条件 ～ 地址

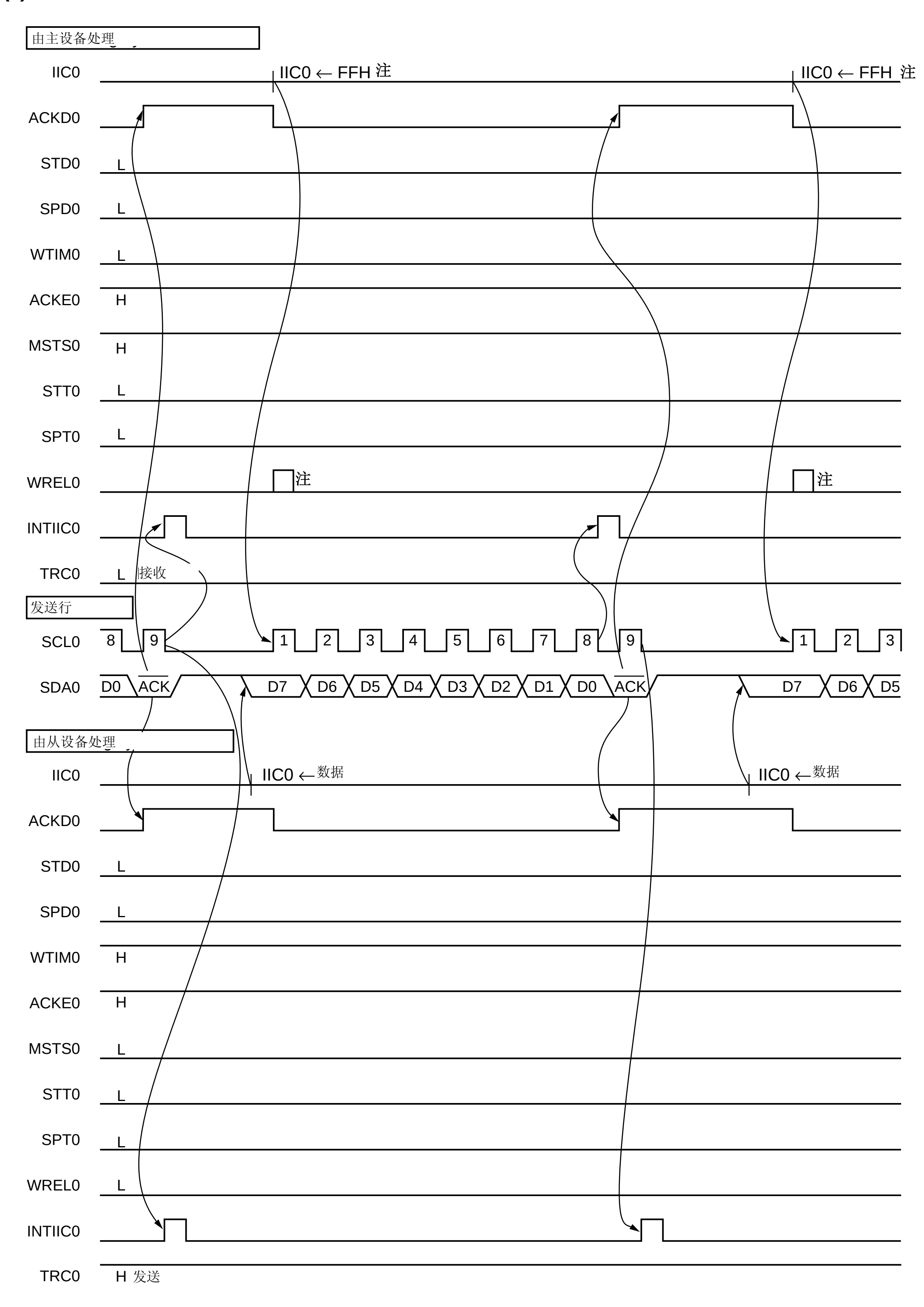



注　　将“FFH”写入 IIC0 或设置 WREL0，可以取消主设备的等待状态。

图 **16-28.** 从设备至主设备通信举例

**(**当主设备选择 **8-**时钟等待，从设备选择 **9-**时钟等待时**) (2/3)**

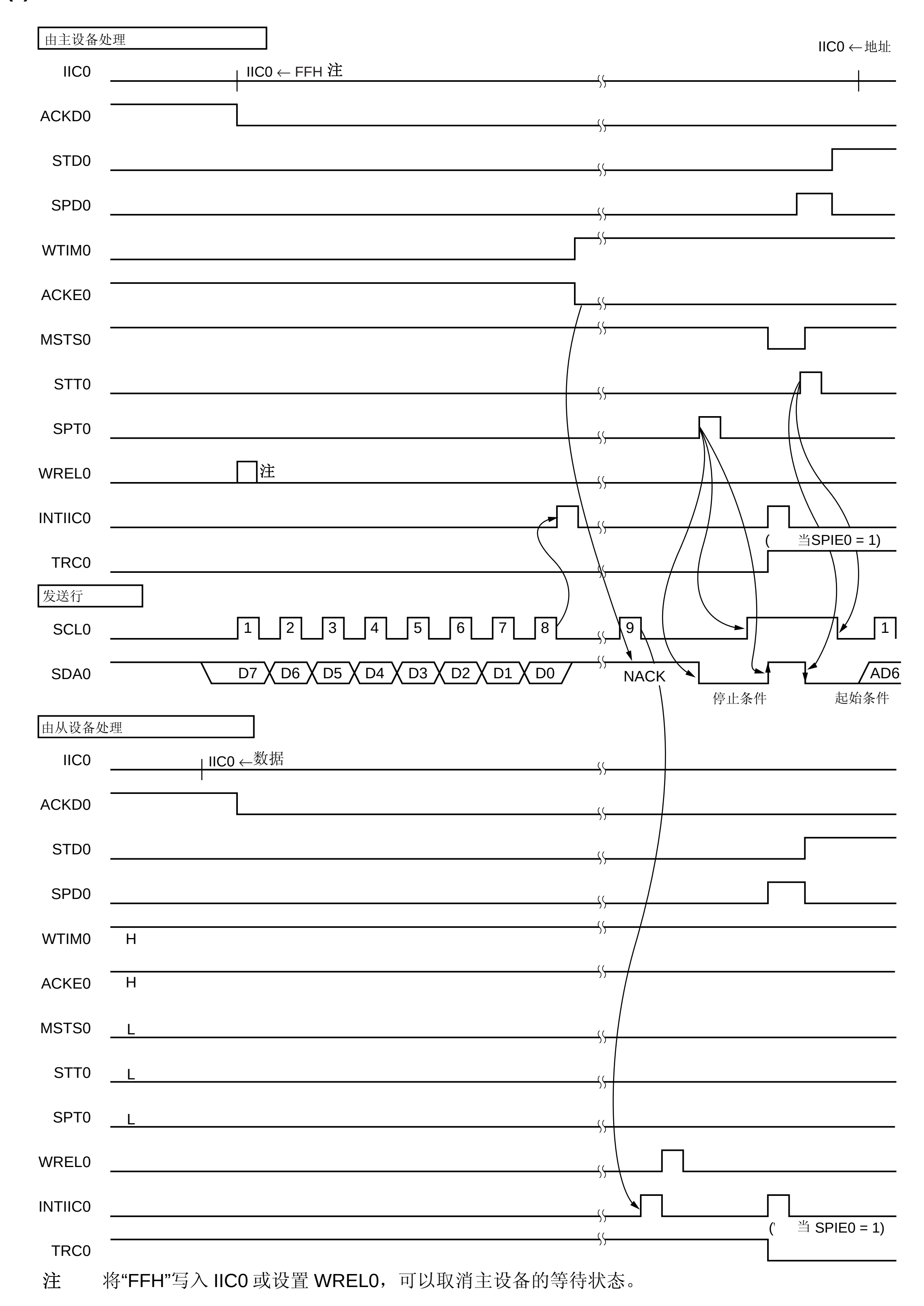
**(2)** 数据




注　　将“FFH”写入 IIC0 或设置 WREL0，可以取消主设备的等待状态。

图 **16-28.** 从设备至主设备通信举例
**(**当主设备选择 **8-**时钟等待，从设备选择 **9-**时钟等待时**) (3/3)**

**(3)** 停止条件




注　　将“FFH”写入 IIC0 或设置 WREL0，可以取消主设备的等待状态。

# 第十七章　乘法器/除法器
# ( 仅 μPD78F0514,  78F0515,  和  78F0515D )

只有 μPD78F0514, 78F0515, 和 78F0515D 提供乘法器/除法器。

## 17.1  乘法器/除法器的功能

乘法器/除法器具有如下功能。

- 16 位 × 16 位= 32 位(乘法)
- 32 位÷ 16 位= 32 位, 16 位余数 (除法)

## 17.2  乘法器/除法器的配置

乘法器/除法器具有如下硬件。

**表 17-1.  乘法器/除法器的配置**

| 项目 | 配置 |
|---|---|
| 寄存器 | 余数寄存器 0 (SDR0)<br>乘/除数寄存器 A0 (MDA0H、MDA0L)<br>乘/除数寄存器 B0 (MDB0) |
| 控制寄存器 | 乘法器/除法器控制寄存器 0 (DMUC0) |

乘法器/除法器的框图如图 17-1 所示。

图 17-1. 乘法器/除法器框图



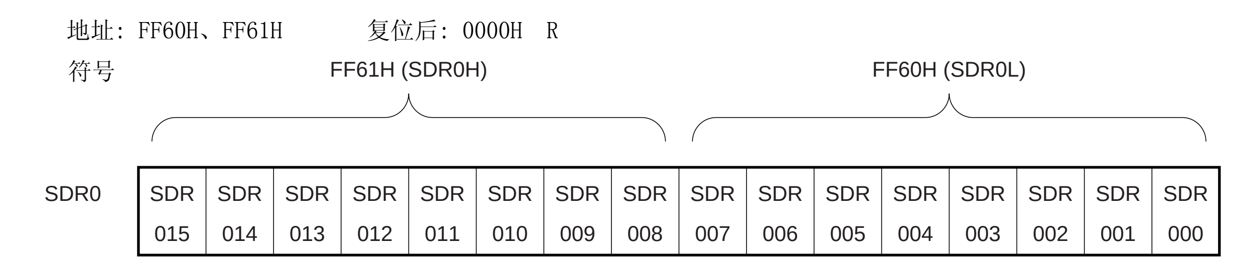
**(1) 余数寄存器 0 (SDR0)**

SDR0 是一个 16 位寄存器，用于存放余数。在乘法模式中该寄存器存储 0，而在除法模式中该寄存器存放操作结果的余数。

可由 8 位或 16 位存储器操作指令读取 SDR0。

复位信号的输入将 SDR0 清零(0000H)。

**图 17-2. 余数寄存器 0 (SDR0)的格式**



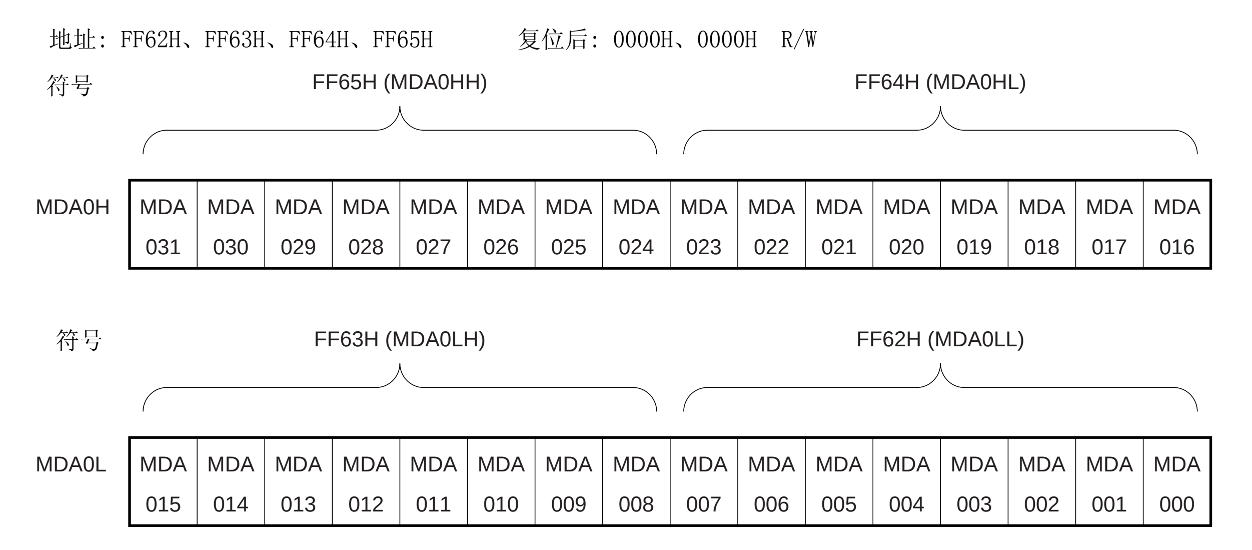

注意事项 **1.** 在操作期间**(**乘法器**/**除法器控制寄存器 **0 (DMUC0)**的第 **7** 位**(DMUE)= 1)**从 **SDR0** 中读取的值不确定。

**2.** 当启动操作时**(**当 **DMUE=1** 时**)**，**SDR0** 被复位。

**(2) 乘/除数寄存器 A0 (MDA0H、MDA0L)**

MDA0 是一个 32 位寄存器，用于在乘法模式中设置 16 位乘数 A 和在除法模式中设置一个 32 位被除数，并存储 32 位的操作结果 (高 16 位: MDA0H、低 16 位: MDA0L)。

**图 17-3. 乘/除数寄存器 A0 (MDA0H、 MDA0L)的格式**




注意事项 **1.** 在乘法模式中**(**乘法器**/**除法器控制寄存器 **0 (DMUC0) = 81H)**启动操作时，**MDA0H** 被清零。

**2.** 在操作期间**(**乘法器**/**除法器控制寄存器 **0 (DMUC0)**的第 **7** 位**(DMUE)= 1)**不要修改 **MDA0** 的值。此时即使执行修改操作，结果也不确定。

**3.** 在操作期间**(**当 **DMUE = 1)**从 **MDA0** 读取的值不确定。

在执行操作时 MDA0 的功能如下。

**表 17-2. 操作执行期间 MDA0 的功能**

| DMUSEL0 | 操作模式 | 设置 | 操作结果 |
|---|---|---|---|
| 0 | 除法模式 | 被除数 | 除法结果 (商) |
| 1 | 乘法模式 | 高 16 位： 0<br>低 16 位： 乘数 A | 乘法结果 (乘积) |

备注 DMUSEL0：乘法器/除法器控制寄存器 0(DMUC0)的第 0 位。

执行乘法操作期间该寄存器的配置与执行除法操作时的不相同，如下所示。

- 乘法操作期间的寄存器配置

  <乘数 A> <乘数 B> <乘积>

  MDA0(15 ～ 0 位) × MDB0(15 ～ 0 位) = MDA0( 31 ～ 0 位)

- 除法操作期间的寄存器配置

  <被除数> <除数> <商> <余数>

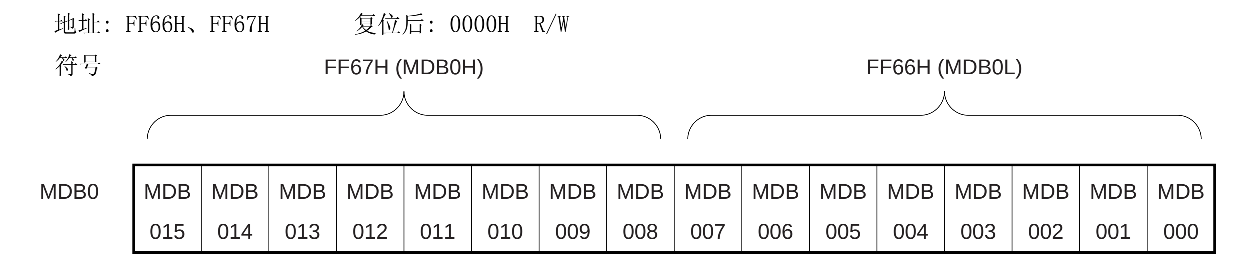
  MDA0(31 ～ 0 位) ÷ MDB0(15 ～ 0 位) = MDA0(31 ～ 0 位)… SDR0(15 ～ 0 位)

当乘法器/除法器控制寄存器 0(DMUC0)的第 7 位(DMUE)=1 时，一旦输入时钟，MDA0 马上获取计算结果。
可由 8 位或 16 位存储器操作指令设置 MDA0H 和 MDA0L。
复位信号的输入将 MDA0H 和 MDA0L 清零(0000H)。

**(3) 乘/除数寄存器 B0 (MDB0)**

MDB0 用于存储乘法模式中的 16 位乘数 B 和除法模式中的 16 位除数。
可由 8 位或 16 位存储器操作指令设置 MDB0。
复位信号的输入将 MDB0 清零(0000H)。

**图 17-4. 乘/除数寄存器 B0 (MDB0)的格式**




注意事项 **1.** **在操作期间(乘法器/除法器控制寄存器 0 (DMUC0)的第 7 位(DMUE)= 1)不要修改 MDB0 的值。此时即使执行修改操作，结果也不确定。**

**2.** **在除法模式中不要对 MDB0 清零(0000H)。如果清零，则存储在 MDA0 与 SDR0 中的操作结果将不确定。**

## 17.3 控制乘法器/除法器的寄存器

乘法器/除法器由乘法器/除法器控制寄存器 0(DMUC0)控制。

**(1) 乘法器/除法器控制寄存器 0 (DMUC0)**

DMUC0 是一个 8 位寄存器，用于控制乘法器/除法器的操作。
可用 1 位或 8 位存储器操作指令设置 DMUC0。
复位信号的输入将 DMUC0 清零(00H)。

**图 17-5. 乘法器/除法器控制寄存器 0 (DMUC0)的格式**

地址：FF68H　　复位后：00H　R/W

| 符号 | <7> | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| DMUC0 | DMUE | 0 | 0 | 0 | 0 | 0 | 0 | DMUSEL0 |

| DMUE [注] | 启动/停止操作 |
|---|---|
| 0 | 停止操作 |
| 1 | 启动操作 |

| DMUSEL0 | 操作模式(乘法/除法)的选择 |
|---|---|
| 0 | 除法模式 |
| 1 | 乘法模式 |

注　当 DMUE =1 时，启动操作。操作结束后 DMUE 被自动清零。

**注意事项**
1. **如果操作期间(当 DMUE = 1)DMUE 被清零，则操作结果不确定。如果设置了中断标志，在执行清零指令时操作完成，操作结果可以得到保证。**
2. **在操作期间(当 DMUE = 1)时，不要修改 DMUSEL0 的值。如果修改了，则存放在乘/除数寄存器 A0 (MDA0)和余数寄存器 0 (SDR0)中的操作结果不确定。**
3. **如果在操作期间(当 DMUE = 1)将 DMUE 清零，则停止操作。若要再次执行操作，应对乘/除数寄存器 A0 (MDA0)、乘/除数寄存器 B0 (MDB0)和乘法器/除法器控制寄存器 0 (DMUC0)进行设置，然后启动操作(通过设置 DMUE = 1)。**

## 17.4 乘法器/除法器的操作

### 17.4.1 乘法操作

- 初始化设置
  1. 将操作数赋给乘/除数寄存器 A0L (MDA0L)和乘/除数寄存器 B0 (MDB0)。
  2. 设置乘法器/除法器控制寄存器 0 (DMUC0)的第 0 位（DMUSEL0）和第 7 位(DMUE)为 1。启动操作。
- 操作期间
  3. 启动操作后经历 16 个内部时钟时，操作完成(操作期间中间数据存于 MDA0L 和 MDA0H 中，因此读取这些寄存器得到的值不确定)。
- 操作结束
  4. 操作结果存于 MDA0L 和 MDA0H 中。
  5. DMUE 被清零 (操作结束)。
  6. 操作结束后，产生中断请求信号(INTDMU)。
- 下一次操作
  7. 执行下一次乘法操作，要从 **17.4.1 乘法操作**中的初始化设置开始。
  8. 执行下一次除法操作，要从 **17.4.2 除法操作**中的初始化设置开始。



**图 17-6.** 乘法操作 **(00DAH × 0093H)** 时序图

### 17.4.2 除法操作

- 初始化设置

1. 将操作数赋给乘/除数寄存器 A0(MDA0L 和 MDA0H)和乘/除数寄存器 B0(MDB0)。
2. 分别设置乘法器/除法器控制寄存器 0(DMUC0)的第 0(DMUSEL0)和第 7 位(DMUE)为 0 和 1。启动操作。

- 操作期间

3. 启动操作后经历 32 个内部时钟时，操作完成(操作期间中间数据存于 MDA0L 和 MDA0H 以及余数寄存器 0(SDR0)中，因此读取这些寄存器得到的值不确定)。

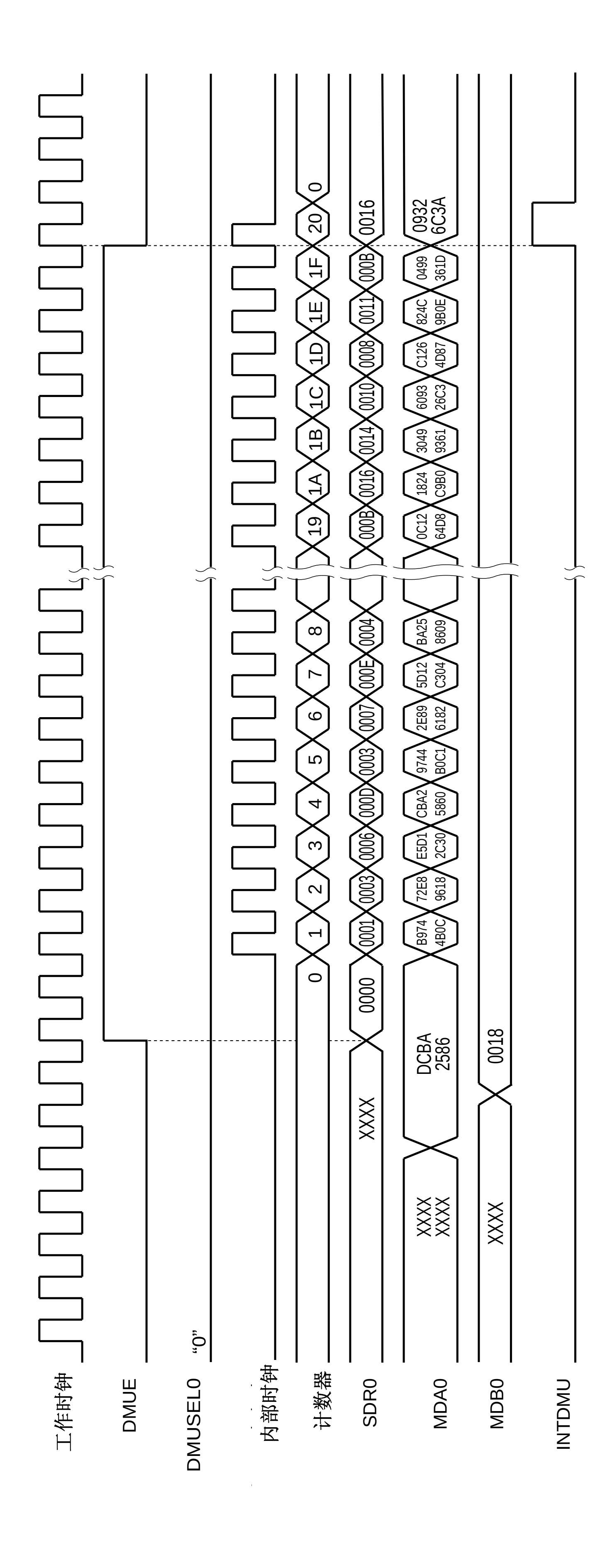
- 操作结束

4. 操作结果存于 MDA0L、MDA0H 和 SDR0 中。
5. DMUE 被清零(操作结束)。
6. 操作结束后，产生中断请求信号(INTDMU)。

- 下一次操作

7. 执行下一次乘法操作，要从 **17.4.1 乘法操作**中的初始化设置开始。
8. 执行下一次除法操作，要从 **17.4.2 除法操作**中的初始化设置开始。

图 17-7．除法操作 (DCBA2586H ÷ 0018H) 的时序图



# 第十八章　中断功能

## 18.1 中断功能的类型

可分为以下两种类型：

**(1) 可屏蔽中断**

这类中断可进行屏蔽控制。通过设置优先级指定标志寄存器(PR0L, PR0H, PR1L, PR1H)将可屏蔽中断分为高优先级中断组和低优先级中断组。

支持中断嵌套，即在低优先级的中断服务子程序中可以响应高优先级的中断请求。如果同时产生两个或两个以上具有相同优先级的中断请求，则根据向量中断服务的优先级进行处理。优先级顺序，请参见**表 18-1**。

产生一个退出待机信号，并退出 STOP 和 HALT 模式。

可屏蔽中断包括外部中断请求和内部中断请求。

- 44 引脚产品　外部： 7， 内部： 16
- 48 引脚产品　外部： 8， 内部： 16

**(2) 软件中断**

这是通过执行 BRK 指令产生的一类向量中断。即使禁止中断时也可以响应这类中断。软件中断不受中断优先级控制。

## 18.2 中断源及配置

78K0/KC2 系列中，44 引脚产品共有 24 种中断源，48 引脚产品共有 25 种中断源，包括可屏蔽的中断和软件中断。另外，还具有 4 种复位源。(参见**表 18-1**)。

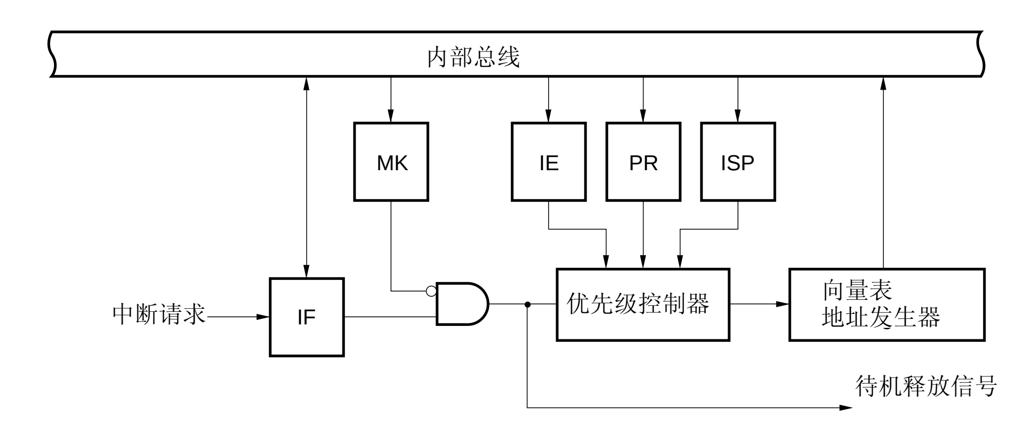
表 18-1. 中断源列表

| 中断类型 | 默认优先级[注 1] | 中断源 名称 | 中断源 触发器 | 内部/外部 | 向量表地址 | 基本配置类型[注 2] |
|---|---|---|---|---|---|---|
| 可屏蔽的 | 0 | INTLVI | 低压检测[注 3] | 内部 | 0004H | (A) |
| | 1 | INTP0 | 引脚输入脉冲沿检测 | 外部 | 0006H | (B) |
| | 2 | INTP1 | | | 0008H | |
| | 3 | INTP2 | | | 000AH | |
| | 4 | INTP3 | | | 000CH | |
| | 5 | INTP4 | | | 000EH | |
| | 6 | INTP5 | | | 0010H | |
| | 7 | INTSRE6 | UART6 产生接收错误 | 内部 | 0012H | (A) |
| | 8 | INTSR6 | UART6 接收结束 | | 0014H | |
| | 9 | INTST6 | UART6 发送结束 | | 0016H | |
| | 10 | INTCSI10/<br>INTST0 | CSI10 通信结束/ UART0 通信结束 | | 0018H | |
| | 11 | INTTMH1 | TMH1 与 CMP01 匹配<br>(指定比较寄存器) | | 001AH | |
| | 12 | INTTMH0 | TMH0 与 CMP00 匹配<br>(指定比较寄存器) | | 001CH | |
| | 13 | INTTM50 | TM50 与 CR50 匹配<br>(指定比较寄存器) | | 001EH | |
| | 14 | INTTM000 | TM00 与 CR000 匹配<br>(指定比较寄存器)，<br>TI010 引脚有效沿检测<br>(指定捕捉寄存器) | | 0020H | |
| | 15 | INTTM010 | TM00 与 CR010 匹配<br>(指定比较寄存器)，<br>TI000 引脚有效沿检测<br>(指定捕捉寄存器) | | 0022H | |
| | 16 | INTAD | A/D 转换结束 | | 0024H | |
| | 17 | INTSR0 | UART0 接收结束或产生接收错误 | | 0026H | |
| | 18 | INTWTI | 钟表定时器参考时间间隔信号 | | 0028H | |
| | 19 | INTTM51[注 4] | TM51 与 CR51 匹配<br>(指定比较寄存器) | | 002AH | |
| | 20 | INTKR | 按键中断检测 | 外部 | 002CH | (C) |
| | 21 | INTWT | 钟表定时器溢出 | 内部 | 002EH | (A) |
| | 22 | INTP6[注 5] | 引脚输入脉冲沿检测 | 外部 | 0030H | (B) |
| | 23 | INTIIC0/<br>INTDMU[注 6] | IIC0 通信结束/乘法/除法操作结束 | 内部 | 0034H | (A) |
| 软件 | – | BRK | 执行 BRK 指令 | – | 003EH | (D) |
| 复位 | – | RESET | 复位输入 | – | 0000H | – |
| | | POC | 上电清零 | | | |
| | | LVI | 低压检测[注 7] | | | |
| | | WDT | WDT 溢出 | | | |

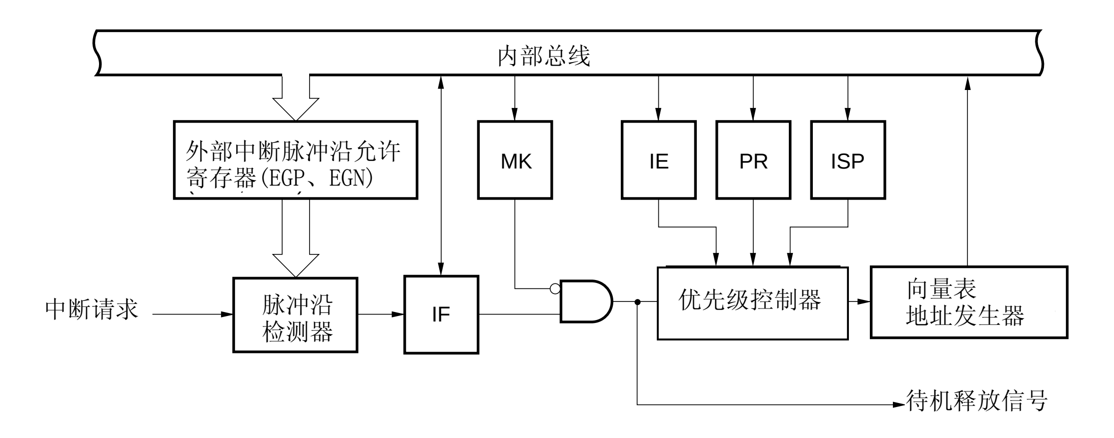
注　1. 当两个或两个以上可屏蔽中断同时产生时，采用默认的优先级来决定向量中断的处理顺序。优先级 0 的优先级别最高，而优先级 23 的优先级别最低。
2. 基本配置类型 (A) ~ (D) 与图 18-1 中 (A) ~ (D) 相对应。
3. 当低电压检测寄存器 (LVIM) 的第 1 位 (LVIMD) 被清零时。
<R> 4. 当 8 位定时器/事件计数器 51 用于载波发生器模式时，中断源为 INTTM5H1 (参看图 **8-13 传输时序**).
5. 中断源 INTP6 仅在 48 引脚产品中可用。
6. 中断源 INTDMU 仅在 μPD78F0514, 78F0515, 和 78F0515D 中可用。
7. 当低电压检测寄存器 (LVIM) 的第 1 位 (LVIMD) 设置为 1 时。

**图 18-1. 中断功能的基本配置 (1/2)**

**(A)** 内部可屏蔽中断



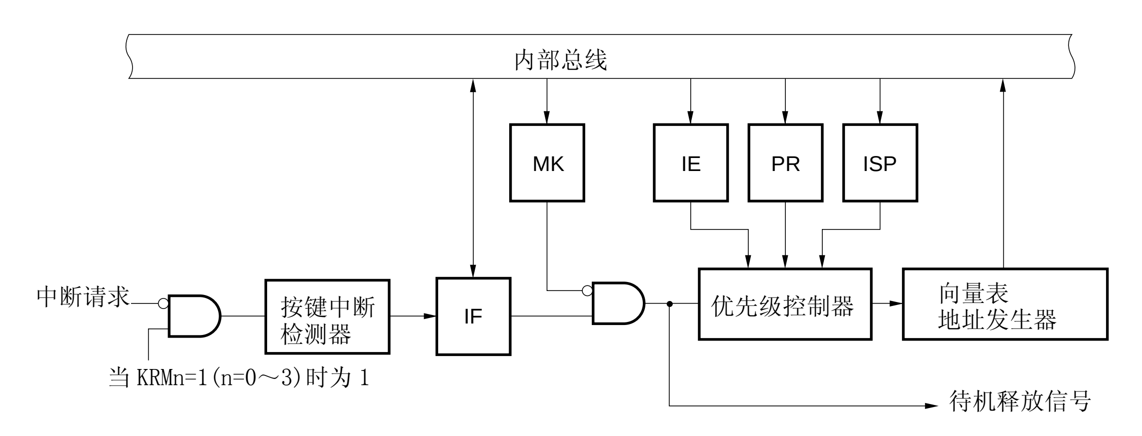

**(B)** 外部可屏蔽中断 **(INTP0 ～ INTP6)**



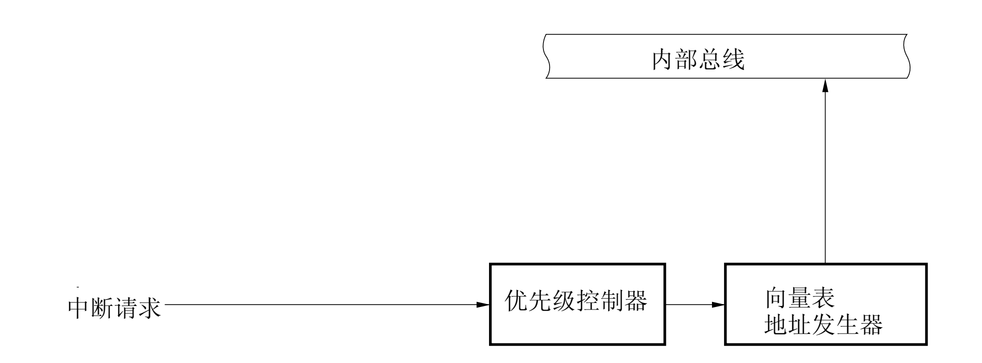

IF：　中断请求标志
IE：　中断允许标志
ISP：　服务优先级标志
MK：　中断屏蔽标志
PR：　优先级指定标志

图 18-1. 中断功能的基本配置 (2/2)

(C) 外部可屏蔽中断 (INTKR)




(D) 软件中断




IF： 中断请求标志
IE： 中断允许标志
ISP： 服务优先级标志
MK： 中断屏蔽标志
PR： 优先级指定标志
KRM：按键返回模式寄存器

## 18.3 控制中断功能的寄存器

以下 6 种寄存器用于控制中断功能。

- 中断请求标志寄存器 (IF0L、IF0H、IF1L、IF1H)
- 中断屏蔽标志寄存器 (MK0L、MK0H、MK1L、MK1H)
- 优先级指定标志寄存器 (PR0L、PR0H、PR1L、PR1H)
- 外部中断上升沿使能寄存器 (EGP)
- 外部中断下降沿使能寄存器 (EGN)
- 程序状态字 (PSW)

表 18-2 列出了与中断请求源相对应的中断请求标志、中断屏蔽标志和优先级指定标志。

表 **18-2.** 与中断请求源相对应的标志

<table>
<tr><th rowspan="2">中断源</th><th colspan="2">中断请求标志</th><th></th><th colspan="2">中断屏蔽标志</th><th></th><th colspan="2">优先级指定标志</th><th></th></tr>
<tr><th colspan="2"></th><th>寄存器</th><th colspan="2"></th><th>寄存器</th><th colspan="2"></th><th>寄存器</th></tr>
<tr><td>INTLVI</td><td colspan="2">LVIIF</td><td rowspan="8">IF0L</td><td colspan="2">LVIMK</td><td rowspan="8">MK0L</td><td colspan="2">LVIPR</td><td rowspan="8">PR0L</td></tr>
<tr><td>INTP0</td><td colspan="2">PIF0</td><td colspan="2">PMK0</td><td colspan="2">PPR0</td></tr>
<tr><td>INTP1</td><td colspan="2">PIF1</td><td colspan="2">PMK1</td><td colspan="2">PPR1</td></tr>
<tr><td>INTP2</td><td colspan="2">PIF2</td><td colspan="2">PMK2</td><td colspan="2">PPR2</td></tr>
<tr><td>INTP3</td><td colspan="2">PIF3</td><td colspan="2">PMK3</td><td colspan="2">PPR3</td></tr>
<tr><td>INTP4</td><td colspan="2">PIF4</td><td colspan="2">PMK4</td><td colspan="2">PPR4</td></tr>
<tr><td>INTP5</td><td colspan="2">PIF5</td><td colspan="2">PMK5</td><td colspan="2">PPR5</td></tr>
<tr><td>INTSRE6</td><td colspan="2">SREIF6</td><td colspan="2">SREMK6</td><td colspan="2">SREPR6</td></tr>
<tr><td>INTSR6</td><td colspan="2">SRIF6</td><td rowspan="9">IF0H</td><td colspan="2">SRMK6</td><td rowspan="9">MK0H</td><td colspan="2">SRPR6</td><td rowspan="9">PR0H</td></tr>
<tr><td>INTST6</td><td colspan="2">STIF6</td><td colspan="2">STMK6</td><td colspan="2">STPR6</td></tr>
<tr><td>INTCSI10</td><td>CSIIF10 注 1</td><td rowspan="2">DUALIF0 注 1</td><td>CSIMK10 注 2</td><td rowspan="2">DUALMK0 注 2</td><td>CSIPR10 注 3</td><td rowspan="2">DUALPR0 注 3</td></tr>
<tr><td>INTST0</td><td>STIF0 注 1</td><td>STMK0 注 2</td><td>STPR0 注 3</td></tr>
<tr><td>INTTMH1</td><td colspan="2">TMIFH1</td><td colspan="2">TMMKH1</td><td colspan="2">TMPRH1</td></tr>
<tr><td>INTTMH0</td><td colspan="2">TMIFH0</td><td colspan="2">TMMKH0</td><td colspan="2">TMPRH0</td></tr>
<tr><td>INTTM50</td><td colspan="2">TMIF50</td><td colspan="2">TMMK50</td><td colspan="2">TMPR50</td></tr>
<tr><td>INTTM000</td><td colspan="2">TMIF000</td><td colspan="2">TMMK000</td><td colspan="2">TMPR000</td></tr>
<tr><td>INTTM010</td><td colspan="2">TMIF010</td><td colspan="2">TMMK010</td><td colspan="2">TMPR010</td></tr>
<tr><td>INTAD</td><td colspan="2">ADIF</td><td rowspan="7">IF1L</td><td colspan="2">ADMK</td><td rowspan="7">MK1L</td><td colspan="2">ADPR</td><td rowspan="7">PR1L</td></tr>
<tr><td>INTSR0</td><td colspan="2">SRIF0</td><td colspan="2">SRMK0</td><td colspan="2">SRPR0</td></tr>
<tr><td>INTWTI</td><td colspan="2">WTIIF</td><td colspan="2">WTIMK</td><td colspan="2">WTIPR</td></tr>
<tr><td>INTTM51[注 4]</td><td colspan="2">TMIF51</td><td colspan="2">TMMK51</td><td colspan="2">TMPR51</td></tr>
<tr><td>INTKR</td><td colspan="2">KRIF</td><td colspan="2">KRMK</td><td colspan="2">KRPR</td></tr>
<tr><td>INTWT</td><td colspan="2">WTIF</td><td colspan="2">WTMK</td><td colspan="2">WTPR</td></tr>
<tr><td>INTP6[注 5]</td><td colspan="2">PIF6[注 5]</td><td colspan="2">PMK6[注 5]</td><td colspan="2">PPR6[注 5]</td></tr>
<tr><td>INTIIC0[注 6]</td><td colspan="2">IICIF0[注 8]</td><td rowspan="2">IF1H</td><td colspan="2">IICMK0[注 9]</td><td rowspan="2">MK1H</td><td colspan="2">IICPR0[注 10]</td><td rowspan="2">PR1H</td></tr>
<tr><td>INTDMU[注 6，7]</td><td colspan="2">DMUIF[注 7，8]</td><td colspan="2">DMUMK[注 7，9]</td><td colspan="2">DMUPR[注 7，10]</td></tr>
</table>

注　**1.** 如果产生中断源 INTCSI10 或 INTST0 中的任意一种，则将 IF0H 的第 2 位设置为(1) 。

**2.** MK0H 的第 2 位支持 INTCSI10 和 INTST0 两种中断源。

**3.** PR0H 的第 2 位支持中断源 INTCSI10 和 INTST0。

<R> **4.** 当 8 位定时器/事件计数器 51 用于载波发生器模式时，中断源为 INTTM5H1 (参看图 **8-13 传输时序**)。

**5.** 仅 48 引脚产品。

<R> **6.** 不要同时使用串行接口 IIC0 和乘法/除法器，因为串行接口 IIC0 和乘法/除法器的中断请求源对应的标志都支持这些中断请求源。如果操作串行接口 IIC0 的软件是用 C 语言编译器 CC78K0 开发的，不要在 PM+的 GUI 中选择复选框“使用乘法器/除法器”。

**7.** 仅 $\mu$PD78F0514, 78F0515, 和 78F0515D 。

<R> **8.** 如果产生中断源 INTIIC0 或 INTDMU ， 则 IF1H 的第 0 位设定为 (1)。

<R> **9.** MK1H 的第 0 位支持中断源 INTIIC0 和 INTDMU。

<R> **10.** PR1H 的第 0 位支持中断源 INTIIC0 和 INTDMU。

**(1) 中断请求标志寄存器 (IF0L, IF0H, IF1L, IF1H)**

当产生相关的中断请求或执行指令时，这些中断请求标志被置 1。当执行的指令是响应中断请求或复位输入时，这些标志被清零。

当响应中断时，中断请求标志自动清零，然后进入中断服务程序。

可由 1 位或 8 位存储器操作指令设置 IF0L、IF0H、IF1L 和 IF1H。当 IF0L 与 IF0H、IF1L 与 IF1H 组合起来形成 16 位寄存器 IF0 和 IF1 时，可用 16 位存储器操作指令设置这些寄存器。

复位信号的输入将这些寄存器清零(00H) 。

**图 18-2. 中断请求标志寄存器 (IF0L, IF0H, IF1L, IF1H) 的格式**

地址: FFE0H 复位后: 00H R/W

| 符号 | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| IF0L | SREIF6 | PIF5 | PIF4 | PIF3 | PIF2 | PIF1 | PIF0 | LVIIF |

地址: FFE1H 复位后: 00H R/W

| 符号 | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| IF0H | TMIF010 | TMIF000 | TMIF50 | TMIFH0 | TMIFH1 | DUALIF0<br>CSIIF10<br>STIF0 | STIF6 | SRIF6 |

地址: FFE2H 复位后: 00H R/W

| 符号 | 7 | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| IF1L | 0 | PIF6[注 1] | WTIF | KRIF | TMIF51 | WTIIF | SRIF0 | ADIF |

地址: FFE3H 复位后: 00H R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| IF1H | 0 | 0 | 0 | 0 | 0 | 0 | 0 | IICIF0<br>DMUIF[注 2] |

| XXIFX | 中断请求标志 |
|---|---|
| 0 | 没有产生中断请求信号 |
| 1 | 产生中断请求，中断请求状态 |

注 **1.** 仅 48 引脚产品。

**2.** 仅μPD78F0514, 78F0515, 和 78F0515D 。

注意事项 **1.** 对 **44** 引脚产品，必须对 **IF1L** 的第 **6** 和 **7** 位清零。

对 **48** 引脚产品，必须对 **IF1L** 的第 **7** 位清零。

**2.** 必须对 **1F1L** 第 **7** 位和 **IF1H** 的第 **1** ～ **7** 位清零。

**3.** 当退出待机模式要使用定时器、串行接口或 **A/D** 转换器时，在将中断请求标志清零后对这些部件操作一次。噪音可能会干扰中断请求标志。

注意事项　**4.** 修改中断请求标志寄存器中的标志时，使用一位存储器操作指令 **(CLR1)** 。当用 **C** 语言描述时,由于编译后的汇编指令必须是一位存储器操作指令**(CLR1),**所以应该使用一位操作指令，如**"IF0L.0 = 0;"** 或 **"_asm("clr1 IF0L, 0");"**。

如果一条 **8** 位存储器操作指令如**"IF0L &= 0xfe;",**用 **C** 语言描述,则编译后将被转换为三条汇编指令：

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

在这个例子中，当处于**"mov a, IF0L"**和 **"mov IF0L, a"**之间的时序时，即使同一中断请求标志寄存器 **(IF0L)**的另一位请求标志被设置为 **1**，则该请求标志也将被**"mov IF0L, a"**清零。因此在 **C** 语言中使用 **8** 位存储器操作指令时必须小心。

**(2)　中断屏蔽标志寄存器 (MK0L, MK0H, MK1L, MK1H)**

这些中断屏蔽标志用于允许/禁止相关的可屏蔽中断服务。

可由 1 位或 8 位存储器操作指令设置 MK0L、 MK0H、MK1L 和 MK1H。 当 MK0L 与 MK0H、MK1L 与 MK1H 组合起来形成 16 位寄存器 MK0 与 MK1 时，可由 16 位存储器操作指令设置这些寄存器。

复位信号的输入将这些寄存器的内容设置为 FFH。

**图 18-3.  中断屏蔽标志寄存器的格式 (MK0L, MK0H, MK1L, MK1H)**

地址: FFE4H　复位后: FFH　R/W

| 符号 | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| MK0L | SREMK6 | PMK5 | PMK4 | PMK3 | PMK2 | PMK1 | PMK0 | LVIMK |

地址: FFE5H　复位后: FFH　R/W

| 符号 | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| MK0H | TMMK010 | TMMK000 | TMMK50 | TMMKH0 | TMMKH1 | DUALMK0<br>CSIMK10<br>STMK0 | STMK6 | SRMK6 |

地址: FFE6H　复位后: FFH　R/W

| 符号 | 7 | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| MK1L | 1 | PMK6[注 1] | WTMK | KRMK | TMMK51 | WTIMK | SRMK0 | ADMK |

地址: FFE7H　复位后: FFH　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| MK1H | 1 | 1 | 1 | 1 | 1 | 1 | 1 | IICMK0<br>DMUMK[注 2] |

| XXMKX | 中断服务控制 |
|---|---|
| 0 | 允许中断服务 |
| 1 | 禁止中断服务 |

注　**1.** 仅 48 引脚产品。

**2.** 仅μPD78F0514, 78F0515, 和 78F0515D 。

**注意事项　1.　对 44 引脚产品，必须将 MK1L 的第 6 和 7 位置 1。**

**对 48 引脚产品，必须将 MK1L 的第 7 位置 1。**

**2.　必须对 MK1H 的第 1 位置 1。**

**(3) 优先级指定标志寄存器(PR0L, PR0H, PR1L, PR1H)**

这些优先级指定标志寄存器用于设置相关的可屏蔽中断优先级次序。

可由 1 位或 8 位存储器操作指令设置 PR0L、 PR0H、 PR1L 和 PR1H 。当 PR0L 与 PR0H、PR1L 与 PR1H 组合起来形成 16 位寄存器 PR0 与 PR1 时，可由 16 位存储器操作指令设置这些寄存器。

复位信号的输入将这些寄存器内容设置为 FFH。

**图 18-4. 优先级指定标志寄存器 (PR0L, PR0H, PR1L, PR1H) 的格式**

地址：FFE8H　复位后：FFH　R/W

| 符号 | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PR0L | SREPR6 | PPR5 | PPR4 | PPR3 | PPR2 | PPR1 | PPR0 | LVIPR |

地址：FFE9H　复位后：FFH　R/W

| 符号 | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PR0H | TMPR010 | TMPR000 | TMPR50 | TMPRH0 | TMPRH1 | DUALPR0<br>CSIPR10<br>STPR0 | STPR6 | SRPR6 |

地址：FFEAH　复位后：FFH　R/W

| 符号 | 7 | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PR1L | 1 | PPR6[注 1] | WTPR | KRPR | TMPR51 | WTIPR | SRPR0 | ADPR |

地址：FFEBH　复位后：FFH　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| PR1H | 1 | 1 | 1 | 1 | 1 | 1 | 1 | IICPR0<br>DMUPR[注 2] |

| XXPRX | 优先级选择 |
|---|---|
| 0 | 高优先级 |
| 1 | 低优先级 |

注　**1.** 仅 48 引脚产品。

　　**2.** 仅μPD78F0514, 78F0515, 和 78F0515D 。

**注意事项　1. 对 44 引脚产品，必须将 PR1L 的第 6 和 7 位置 1。**

**对 48 引脚产品，必须将 PR1L 的第 7 位置 1。**

**2. 必须对 PR1H 的第 1 位置 1。**

**(4)　外部中断上升沿允许寄存器(EGP)、 外部中断下降沿允许寄存器(EGN)**

这两个寄存器用于指定 INTP0 到 INTP6 的有效沿。

可由 1 位或 8 位存储器操作指令设置 EGP 和 EGN。

复位信号的输入将这些寄存器清零 (00H)。

**图 18-5.外部中断上升沿允许寄存器(EGP) 和外部中断下降沿允许寄存器(EGN) 的格式**

地址: FF48H　复位后: 00H　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| EGP | 0 | EGP6[注] | EGP5 | EGP4 | EGP3 | EGP2 | EGP1 | EGP0 |

地址: FF49H　复位后: 00H　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| EGN | 0 | EGN6[注] | EGN5 | EGN4 | EGN3 | EGN2 | EGN1 | EGN0 |

| EGPn | EGNn | INTPn 引脚有效沿的选择 (n = 0 ～ 6) |
|---|---|---|
| 0 | 0 | 禁止脉冲沿检测 |
| 0 | 1 | 下降沿 |
| 1 | 0 | 上升沿 |
| 1 | 1 | 兼有上升沿和下降沿 |

注　　仅 48 引脚产品。

**注意事项　对 44 引脚产品，必须对 EGP 和 EGN 的第 6 和 7 位清零。**
**对 48 引脚产品，必须对 EGP 和 EGN 的第 7 位清零。**

表 18-3 显示了与 EGPn 和 EGNn 相关的端口。

**表 18-3.　与 EGPn 和 EGNn 相关的端口**

| 检测允许寄存器 | | 脉冲沿检测端口 | 中断请求信号 |
|---|---|---|---|
| EGP0 | EGN0 | P120 | INTP0 |
| EGP1 | EGN1 | P30 | INTP1 |
| EGP2 | EGN2 | P31 | INTP2 |
| EGP3 | EGN3 | P32 | INTP3 |
| EGP4 | EGN4 | P33 | INTP4 |
| EGP5 | EGN5 | P16 | INTP5 |
| EGP6[注] | EGN6[注] | P140 | INTP6 |

注　　仅 48 引脚产品。

**注意事项　在从外部中断功能切换到端口功能时可能会检测到脉冲沿，因此通过将 EGPn 和 EGNn 清零可选择端口模式。**

备注　　n = 0 ～ 6

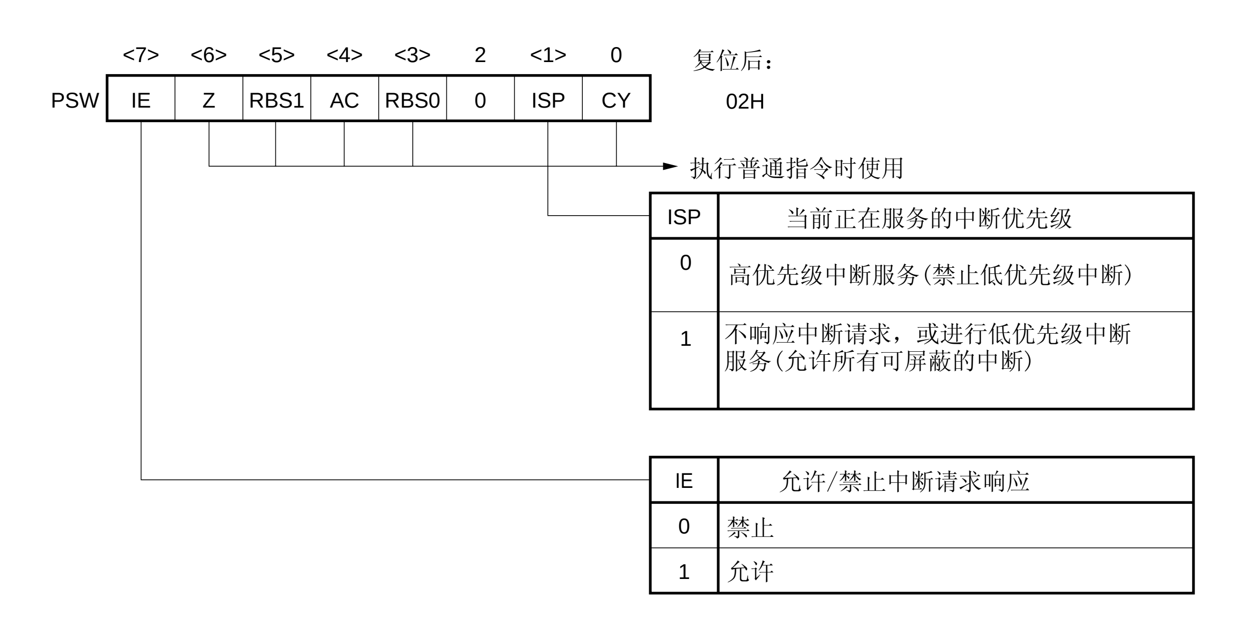
**(5) 程序状态字(PSW)**

PSW 用于保存指令执行结果和中断请求的当前状态。在 PSW 中包含 IE 标志（设置允许/禁止可屏蔽中断）和 ISP 标志（控制中断嵌套）。

除了 8 位读/写操作指令，还可使用位操作指令和专用指令（EI 和 DI）对该寄存器进行操作。在响应向量中断请求时，如果执行 BRK 指令，则将 PSW 的内容自动保存到堆栈中，并且将 IE 标志复位为 0。如果响应可屏蔽中断请求，则将被响应中断的优先级指定标志的内容转移到 ISP 标志中。执行 PUSH PSW 指令将 PSW 的内容保存到堆栈中。而执行 RETI、RETB 和 POP PSW 指令可将这些内容从堆栈中恢复。

复位信号的输入将 PSW 设置为 02H。

**图 18-6. 程序状态字的格式**



## 18.4 中断服务操作

### 18.4.1 可屏蔽的中断响应

当中断请求标志=1 且与该中断请求相关的屏蔽标志(MK)被清零时，可以响应这个可屏蔽中断请求。如果处于中断允许状态(IE =1)，可以响应向量中断请求。但在一个较高优先级中断请求服务期间（ISP 标志复位为 0），不响应低优先级的中断请求。

从一个可屏蔽中断请求产生到中断服务执行所经历的时间如表 18-4 所示。

中断请求响应时序可参见图 **18-8** 和 **18-9**。

**表 18-4. 从可屏蔽中断产生到执行中断服务所需要的时间**

| | 最短时间 | 最长时间[注] |
|---|---|---|
| 当××PR = 0 | 7 个时钟 | 32 个时钟 |
| 当××PR = 1 | 8 个时钟 | 33 个时钟 |

**注**　如果是在除法指令执行之前产生一个中断请求，则等待时间会更长。

**备注**　1 个时钟： $1/f_{CPU}$ ($f_{CPU}$： CPU 时钟)

如果同时产生两个或两个以上的可屏蔽中断请求，则首先响应优先级指定标志中优先级别高的请求。如果两个或两个以上的中断请求具有相同的优先级别，则首先响应具有最高默认优先级的中断请求。

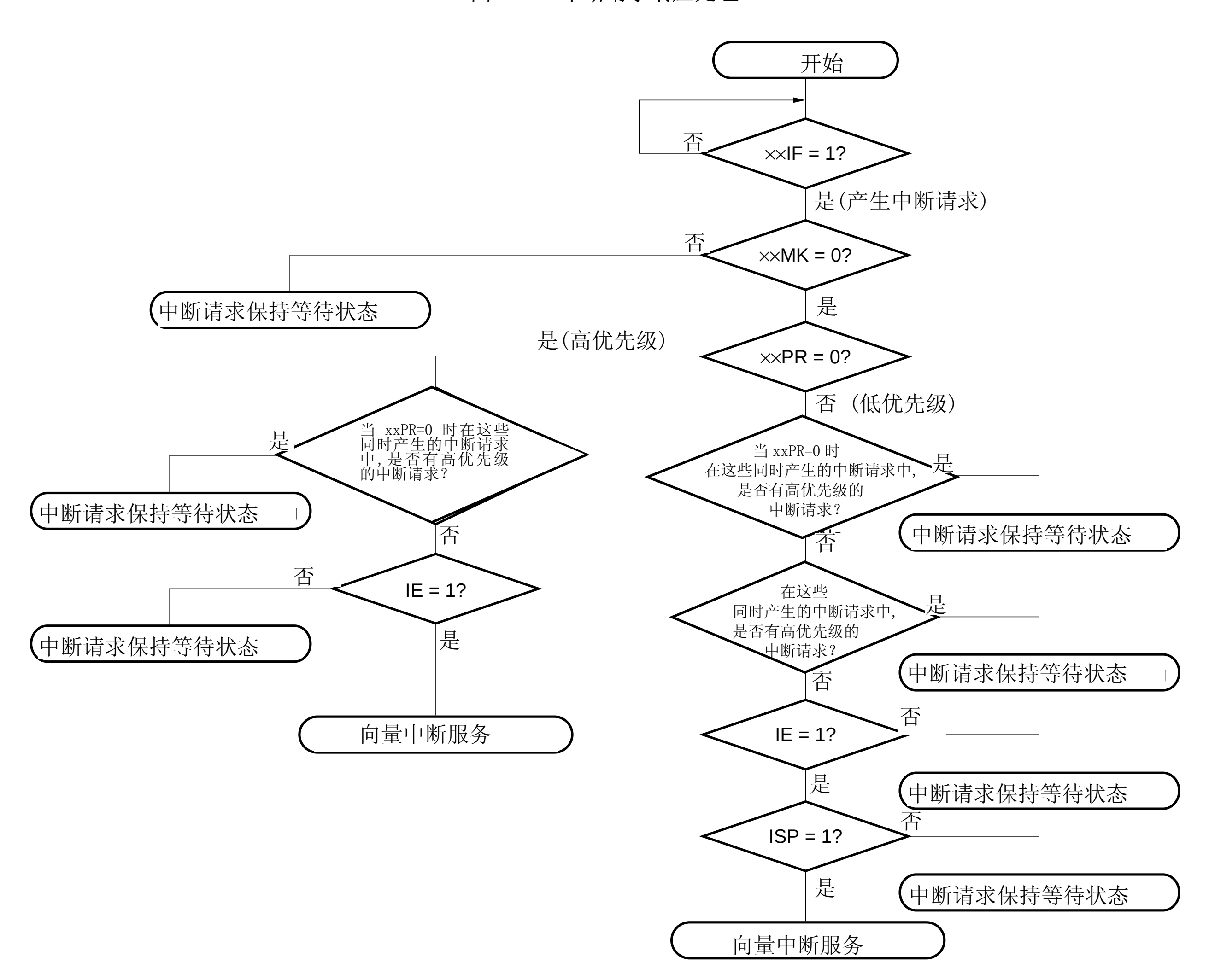
当允许响应中断请求时，处于等待状态的中断请求被响应。

图 18-7 显示了中断请求响应算法。

如果响应了一个可屏蔽中断请求，则将 PSW、PC 的内容依次保存到堆栈中，然后将 IE 标志复位为 0，并将与被响应的中断相关的优先级指定标志的内容传送到 ISP 中。将用于中断请求的向量表数据传送到 PC 中，并转移。

可通过执行 RETI 指令从中断返回。

**图 18-7.  中断请求响应处理**




××IF：  中断请求标志

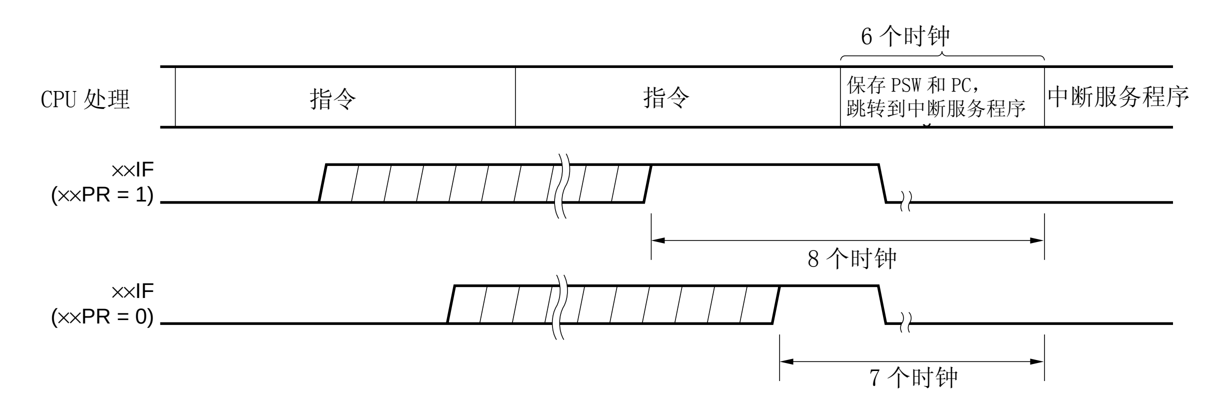
××MK：中断屏蔽标志

××PR：优先级指定标志

IE：    控制可屏蔽中断请求响应的标志 （1 = 允许、 0 = 禁止）

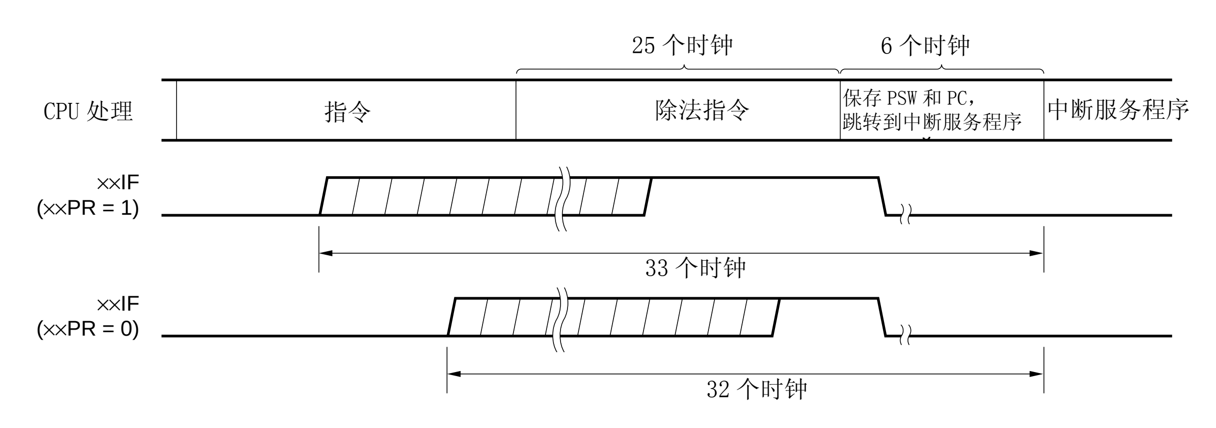
ISP：   指示正在服务的中断优先级别的标志 （0 = 高优先级的中断服务，1 = 没有中断请求被响应，或低优先级的中断服务）

图 **18-8.** 中断请求响应时序 (最短时间)




备注　　1 个时钟： 1/$f_{CPU}$ ($f_{CPU}$：　CPU 时钟)

图 **18-9.** 中断请求响应时序 (最长时间)




备注　　1 个时钟： 1/$f_{CPU}$ ($f_{CPU}$：　CPU 时钟)

### 18.4.2 软件中断请求响应

执行 BRK 指令可响应软件中断。软件中断不能被禁止。

如果响应了一个软件中断请求，则将程序状态字（PSW）和程序计数器（PC）的内容依次保存到堆栈中，然后将 IE 标志复位为 0，并将向量表的内容（003EH、003FH）传送到 PC 中，然后转移。

可通过执行 RETB 指令从软件中断返回。

**注意事项　不能使用 RETI 指令从软件中断返回。**

### 18.4.3 中断嵌套

在执行一个中断服务程序时，又响应了其他中断，这时就产生了中断嵌套。

除非选择允许中断请求响应状态（IE = 1），否则不会产生中断嵌套。在响应一个中断请求时，禁止响应其它中断请求（IE = 0）。因此，如果要允许中断嵌套，必须在中断服务期间执行 EI 指令，将 IE 标志置 1，从而允许响应其它中断请求。

此外，即使允许中断，也不一定允许中断嵌套，这是因为受到中断优先级控制的限制。可使用两类优先级控制方式：默认优先级控制和可编程优先级控制。可编程优先级控制用于中断嵌套。

在中断允许状态中，如果产生的中断请求的优先级与正在服务的中断优先级相等或高于它，则响应该中断请求，从而产生中断嵌套。如果产生的中断请求的优先级低于正在服务的中断优先级，则不响应该中断请求。由于禁止中断或中断请求的优先级别较低，这些不被允许的中断请求处于等待状态。若当前的中断服务已结束，则在执行至少一条主程序指令后才可响应处于等待状态的中断请求。

表 19-5 显示了允许中断嵌套的中断请求之间的关系，图 19-10 为中断嵌套示例。

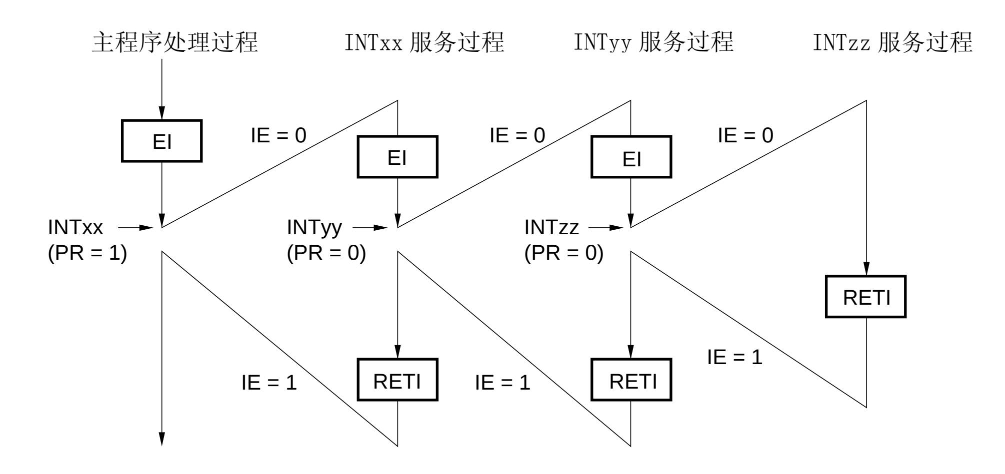
**表 18-5. 中断服务期间允许进行中断嵌套的中断请求之间的关系**

| 中断嵌套 / 正在服务的中断 | | 可屏蔽中断请求 | | | | 软件中断请求 |
|---|---|---|---|---|---|---|
| | | PR = 0 | | PR = 1 | | |
| | | IE = 1 | IE = 0 | IE = 1 | IE = 0 | |
| 可屏蔽的中断 | ISP = 0 | ○ | × | × | × | ○ |
| | ISP = 1 | ○ | × | ○ | × | ○ |
| 软件中断 | | ○ | × | ○ | × | ○ |

**备注**
1. ○：　允许中断嵌套
2. ×：　禁止中断嵌套
3. ISP 和 IE 为 PSW 中的标志位。
   ISP = 0：正在服务一个高优先级中断。
   ISP = 1：无中断请求被响应，或正在服务一个低优先级中断。
   IE = 0：　禁止响应中断请求。
   IE = 1：　允许响应中断请求。
4. PR 为 PR0L、PR0H、PR1L 和 PR1H 中的标志位。
   PR = 0：高优先级
   PR = 1：低优先级

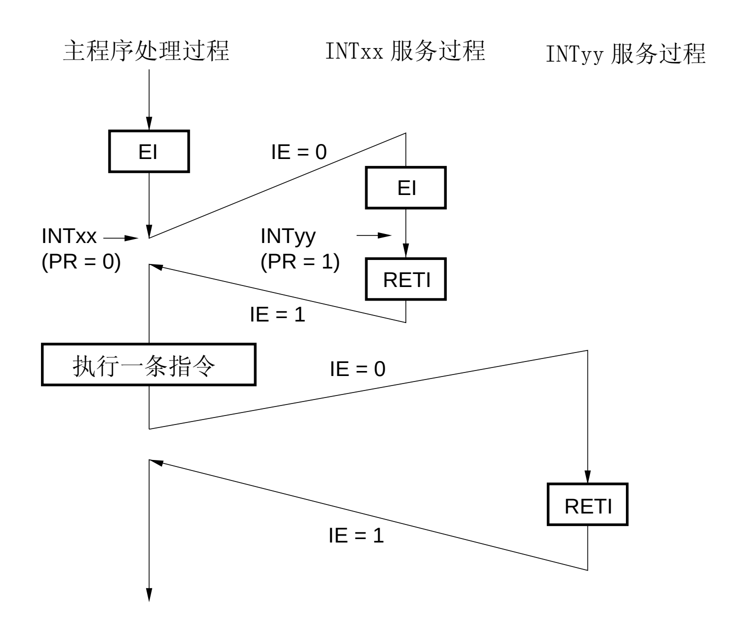
图 **18-10.** 中断嵌套示例 **(1/2)**

例 **1.** 产生两次中断嵌套




在进行 INTxx 中断服务期间，响应了两个中断请求 INTyy 和 INTzz，这时就产生了中断嵌套。在响应每个中断之前，必须先执行 EI 指令才能允许响应中断请求。

例 **2.** 由于优先级控制没有产生中断嵌套

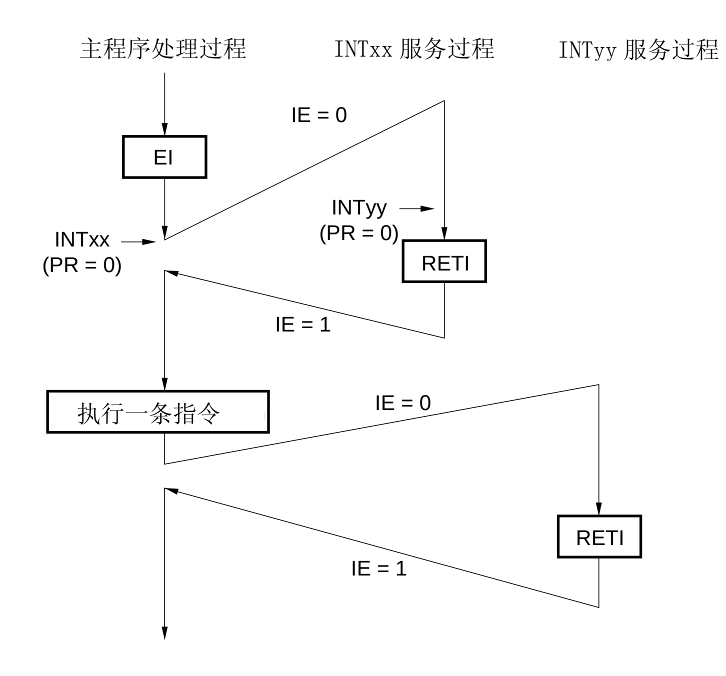



在 INTxx 中断服务期间不响应中断请求 INTyy，因为 INTyy 的优先级低于 INTxx，这样就不会产生中断嵌套。中断请求 INTyy 处于等待状态，在执行一条主程序指令后响应该中断请求。

PR = 0：高优先级
PR = 1：低优先级
IE = 0：禁止响应中断请求

图 **18-10.** 中断嵌套示例 **(2/2)**

例 **3.** 由于不允许中断没有产生中断嵌套




在 INTxx 中断服务期间不允许中断（不执行 EI 指令），因此不响应中断请求 INTyy，这样也不会产生中断嵌套。中断请求 INTyy 处于等待状态，在执行一条主程序指令后响应该中断请求。

PR = 0：高优先级
IE = 0：禁止响应中断请求

### 18.4.4 保持中断请求

在某些指令执行期间，即使出现中断请求，请求响应也要保持等待状态，直到下一条指令执行结束。以下列出这类指令（中断请求保持指令）。

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW. bit, CY
- MOV1 CY, PSW. bit
- AND1 CY, PSW. bit
- OR1 CY, PSW. bit
- XOR1 CY, PSW. bit
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW. bit, $addr16
- BF PSW. bit, $addr16
- BTCLR PSW. bit, $addr16
- EI
- DI
- 用于 IF0L, IF0H, IF1L, IF1H, MK0L, MK0H, MK1L, MK1H, PR0L, PR0H, PR1L, 和 PR1H 寄存器的操作指令。

注意事项 **BRK 指令不属于上述列出的中断请求保持指令。但通过执行 BRK 指令激活的软件中断会将 IE 标志清零。因此，即使在执行 BRK 指令期间产生可屏蔽中断请求，该中断请求也不会被响应。**

图 18-11 显示了处于等待状态的中断请求时序。

**图 18-11. 中断请求保持时序**




备注 **1.** 指令 N： 中断请求保持指令
**2.** 指令 M： 除中断请求保持指令之外的指令
**3.** ××PR（优先级别）的值不会影响××IF（中断请求）的操作。

## 19.1 按键中断功能

可通过设置按键返回模式寄存器(KRM)和向按键中断输入引脚(KR0～KR3)输入一个下降沿产生按键中断 (INTKR)。

**表 19-1. 按键中断检测引脚的分配**

| 标识 | 描述 |
|---|---|
| KRM0 | 以 1 位操作模式控制 KR0 信号 |
| KRM1 | 以 1 位操作模式控制 KR1 信号 |
| KRM2 | 以 1 位操作模式控制 KR2 信号 |
| KRM3 | 以 1 位操作模式控制 KR3 信号 |

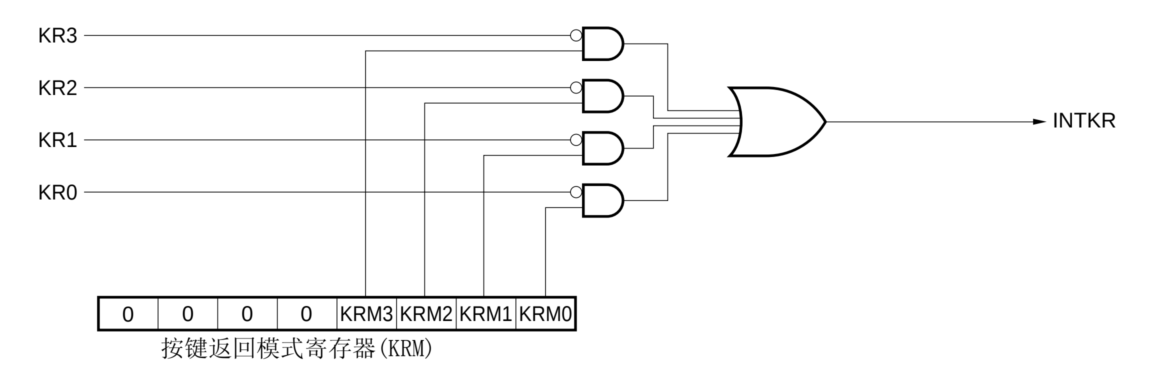
## 19.2 按键中断的配置

按键中断包括以下硬件。

**表 19-2. 按键中断的配置**

| 项目 | 配置 |
|---|---|
| 控制寄存器 | 按键返回模式寄存器(KRM) |

**图 19-1. 按键中断的框图**



## 19.3 控制按键中断的寄存器

**(1) 按键返回模式寄存器 (KRM)**

该寄存器使用 KR0 ～ KR3 信号分别控制 KRM0 ～ KRM3 位。

KRM 可由 1 位或 8 位存储器操作指令设置。

复位信号的产生将 KRM 清零(00H)。

图 **19-2.** 按键返回模式寄存器 **(KRM)** 的格式

地址：FF6EH　　复位后：00H　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| KRM | 0 | 0 | 0 | 0 | KRM3 | KRM2 | KRM1 | KRM0 |

| KRMn | 按键中断模式控制 |
|---|---|
| 0 | 不检测按键中断信号 |
| 1 | 检测按键中断信号 |

注意事项 **1.** 如果 **KRM0** ～ **KRM3** 中任何一位被设置为 **1**，则将上拉电阻寄存器 **7(PU7)**的 **0** ～ **3** 位**(PU70** ～ **PU73)**的相应位设置为 **1**。

**2.** 如果 **KRM** 发生变化，则中断请求标志可能被设置。因此可以先禁止中断，再修改 **KRM**。先将中断请求标志清零，再允许中断。

**3.** 在按键中断模式中没有被使用的位可用作通用端口。

# 第二十章　待机功能

## 20.1 待机功能及配置

### 20.1.1 待机功能

待机功能用于减少系统的工作电流，有以下两种模式。

**(1) HALT 模式**

通过执行 HALT 指令设置 HALT 模式。在 HALT 模式中，CPU 操作时钟停止。如果设置 HALT 模式前，高速系统时钟振荡器、内部高速振荡器、内部低速振荡器或副系统时钟振荡器正在使用，则设置后每种时钟的振荡继续。在此模式中，工作电流不如 STOP 模式中下降得多，但 HALT 模式对于中断请求产生后立即重启操作和频繁进行间断的操作非常有效。

**(2) STOP 模式**

通过执行 STOP 指令设置 STOP 模式。在 STOP 模式中，高速系统时钟振荡器和内部高速振荡器停止操作，整个系统的操作停止，这样 CPU 的工作电流将会大幅下降。

可通过中断请求释放该模式，这样，被中断的操作可以继续执行。由于选择 X1 时钟时，在释放 STOP 模式后需要一段等待时间以确保振荡器振荡稳定，因此如果需要在产生中断请求后立即进行处理，则应选择 HALT 模式。

在这两种模式中，寄存器、标志和数据存储器的内容将会保持进入待机模式前的内容。I/O 端口输出锁存器和输出缓冲器的状态也将被保持。

注意事项 **1. 仅当 CPU 使用主系统时钟时，才能使用 STOP 模式。不能停止副系统时钟的振荡。而当 CPU 使用主系统时钟或副系统时钟时，均可以使用 HALT 模式。**

**2. 当切换到 STOP 模式时，在执行 STOP 指令前必须停止使用主系统时钟的外围硬件的操作。**

**3. 当使用待机功能时，建议采用以下步骤降低 A/D 转换器的操作电流：首先将 A/D 转换器模式寄存器（ADM）的第 7 位（ADCS）和第 0 位（ADCE）清零以停止 A/D 转换操作，然后执行 STOP 指令。**

### 20.1.2 控制待机功能的寄存器

待机功能由以下两个寄存器控制。

- 振荡稳定时间计数器的状态寄存器 （OSTC）
- 振荡稳定时间选择寄存器 （OSTS）

**备注**　有关这些寄存器启动、停止以及时钟选择的情况，可参见 **第五章　时钟发生器。**

**(1) 振荡稳定时间计数器的状态寄存器 (OSTC)**

这是 X1 时钟振荡稳定时间计数器的状态寄存器。如果 CPU 使用 X1 振荡时钟、内部高速振荡时钟或副系统时钟，则可以检测 X1 时钟振荡稳定时间。

可由 1 位或 8 位存储器操作指令读取 OSTC。

复位释放（通过 RESET 输入、POC、LVI 和 WDT）、STOP 指令以及 MSTOP （MOC 寄存器的第 7 位） = 1 可以将 OSTC 清零（00H）。

**图 20-1. 振荡稳定时间计数器的状态寄存器 （OSTC） 的格式**

地址: FFA3H　复位后: 00H　R

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| OSTC | 0 | 0 | 0 | MOST11 | MOST13 | MOST14 | MOST15 | MOST16 |

| MOST11 | MOST13 | MOST14 | MOST15 | MOST16 | 振荡稳定时间的状态 | | |
|---|---|---|---|---|---|---|---|
| | | | | | | fx = 10 MHz | fx = 20 MHz |
| 1 | 0 | 0 | 0 | 0 | $2^{11}$/fx min. | 204.8 μs min. | 102.4 μs min. |
| 1 | 1 | 0 | 0 | 0 | $2^{13}$/fx min. | 819.2 μs min. | 409.6 μs min. |
| 1 | 1 | 1 | 0 | 0 | $2^{14}$/fx min. | 1.64 ms min. | 819.2 μs min. |
| 1 | 1 | 1 | 1 | 0 | $2^{15}$/fx min. | 3.27 ms min. | 1.64 ms min. |
| 1 | 1 | 1 | 1 | 1 | $2^{16}$/fx min. | 6.55 ms min. | 3.27 ms min. |

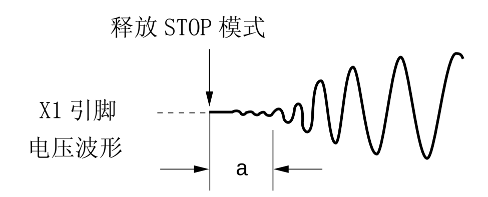
注意事项 **1.** 在经历上述时间后，从 **MOST11** 起各位被设置为 **1**，并保持为 **1**。

**2.** 振荡稳定时间计数器的计数达到 **OSTS** 设置的振荡稳定时间。如果已进入 **STOP** 模式并在 **CPU** 使用内部高速振荡时钟时释放 **STOP** 模式，则可以按以下方式设置振荡稳定时间。

- 预期的 **OSTC** 振荡稳定时间 ≤ **OSTS** 设置的振荡稳定时间

因此需要注意，在释放 **STOP** 模式后，只有通过 **OSTS** 设置的振荡稳定时间期间的状态被设置到 **OSTC**。

**3. X1** 时钟振荡稳定等待时间不包括从释放 **STOP** 模式到时钟振荡启动这段时间（即下图 **"a"** 所示的部分）。




备注　fx: X1 时钟振荡频率。

**(2) 振荡稳定时间选择寄存器(OSTS)**

该寄存器用于选择释放 STOP 模式后 X1 时钟振荡稳定等待时间。

在释放 STOP 模式后且 CPU 使用 X1 时钟时，等待由 OSTS 设置的时间。

当释放 STOP 模式后且 CPU 使用内部高速振荡时钟时，可通过使用 OSTC 确认是否已经历了预期的振荡稳定时间。振荡稳定时间可根据 OSTC 设置的时间来检测。

可由 8 位存储器操作指令设置 OSTS。

复位信号的产生将 OSTS 设置为 05H。

图 **20-2.** 振荡稳定时间选择寄存器 （**OSTS**）的格式

地址: FFA4H　复位后: 05H　R/W

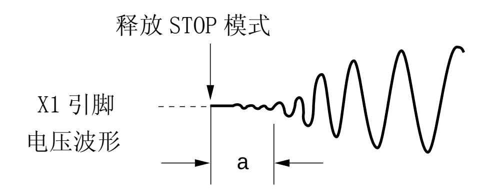
| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| OSTS | 0 | 0 | 0 | 0 | 0 | OSTS2 | OSTS1 | OSTS0 |

| OSTS2 | OSTS1 | OSTS0 | 振荡稳定时间的状态 | | |
|---|---|---|---|---|---|
| | | | | $f_X$ = 10 MHz | $f_X$ = 20 MHz |
| 0 | 0 | 1 | $2^{11}/f_X$ | 204.8 μs | 102.4 μs |
| 0 | 1 | 0 | $2^{13}/f_X$ | 819.2 μs | 409.6 μs |
| 0 | 1 | 1 | $2^{14}/f_X$ | 1.64 ms | 819.2 μs |
| 1 | 0 | 0 | $2^{15}/f_X$ | 3.27 ms | 1.64 ms |
| 1 | 0 | 1 | $2^{16}/f_X$ | 6.55 ms | 3.27 ms |
| 其它 | | | 禁止设置 | | |

注意事项

1. 如果在 **X1** 时钟用作 **CPU** 时钟时要设置 **STOP** 模式，则应在执行 **STOP** 指令之前设置 **OSTS**。
2. 在 **X1** 时钟振荡稳定期间不要修改 **OSTS** 寄存器的值。
3. 振荡稳定时间计数器的计数达到 **OSTS** 设置的振荡稳定时间。如果已进入 **STOP** 模式，并在 **CPU** 使用内部高速振荡时钟时释放 **STOP** 模式，则可以按以下方式设置振荡稳定时间。
    - 预期的 **OSTC** 振荡稳定时间 ≤ **OSTS** 设置的振荡稳定时间

    因此需要注意，在释放 **STOP** 模式后，只有通过 **OSTS** 设置的振荡稳定时间期间的状态被设置到 **OSTC**。
4. **X1** 时钟振荡等待时间不包括从释放 **STOP** 模式到时钟振荡启动这段时间（即下图“**a**”所示的部分）。




备注　$f_X$: X1 时钟振荡频率

## 20.2 待机功能的操作

### 20.2.1 HALT 模式

**(1) HALT 模式**

通过执行 HALT 指令设置 HALT 模式。无论设置前 CPU 使用的是高速系统时钟、内部高速振荡时钟还是副系统时钟，都可以设置 HALT 模式。

HALT 模式中的操作状态如下所示。

**表 20-1. HALT 模式中的操作状态 (1/2)**

<table>
<tr><td colspan="3" rowspan="2">HALT 模式设置<br>项目</td><td colspan="3">当 CPU 使用主系统时钟并执行 HALT 指令时</td></tr>
<tr><td>当CPU使用内部高速振荡时钟 ($f_{RH}$)时</td><td>当 CPU 使用 X1 时钟($f_X$)时</td><td>当 CPU 使用外部主系统时钟 ($f_{EXCLK}$)时</td></tr>
<tr><td colspan="3">系统时钟</td><td colspan="3">停止 CPU 时钟</td></tr>
<tr><td rowspan="6"></td><td rowspan="3">主系统时钟</td><td>$f_{RH}$</td><td>操作继续 (不能停止)</td><td colspan="2">保持设置 HALT 模式前的状态</td></tr>
<tr><td>$f_X$</td><td>保持设置 HALT 模式前的状态</td><td>操作继续 (不能停止)</td><td>保持设置 HALT 模式前的状态</td></tr>
<tr><td>$f_{EXCLK}$</td><td colspan="2">由外部时钟输入进行操作或停止</td><td>操作继续 (不能停止)</td></tr>
<tr><td rowspan="2">副系统时钟</td><td>$f_{XT}$</td><td colspan="3">保持设置 HALT 模式前的状态</td></tr>
<tr><td>$f_{EXCLKS}$</td><td colspan="3">由外部时钟输入进行操作或停止</td></tr>
<tr><td colspan="2">$f_{RL}$</td><td colspan="3">保持设置 HALT 模式前的状态</td></tr>
<tr><td colspan="3">CPU</td><td colspan="3" rowspan="2">操作停止</td></tr>
<tr><td colspan="3">Flash 存储器</td></tr>
<tr><td colspan="3">RAM</td><td colspan="3" rowspan="2">保持设置 HALT 模式前的状态</td></tr>
<tr><td colspan="3">端口（锁存器）</td></tr>
<tr><td colspan="3">16 位定时器/事件计数器 00</td><td colspan="3" rowspan="6">可操作</td></tr>
<tr><td colspan="2" rowspan="2">8 位定时器/事件计数器</td><td>50</td></tr>
<tr><td>51</td></tr>
<tr><td colspan="2" rowspan="2">8 位定时器</td><td>H0</td></tr>
<tr><td>H1</td></tr>
<tr><td colspan="3">钟表定时器</td></tr>
<tr><td colspan="3">看门狗定时器</td><td colspan="3">可操作。当通过可选字节设置“可以由软件停止内部低速振荡器”时，停止看门狗定时器的时钟。</td></tr>
<tr><td colspan="3">时钟输出注 1</td><td colspan="3" rowspan="10">可操作</td></tr>
<tr><td colspan="3">A/D 转换器</td></tr>
<tr><td colspan="2" rowspan="4">串行接口</td><td>UART0</td></tr>
<tr><td>UART6</td></tr>
<tr><td>CSI10</td></tr>
<tr><td>IIC0</td></tr>
<tr><td colspan="3">乘法器/除法器注 2</td></tr>
<tr><td colspan="3">上电清零（POC）功能</td></tr>
<tr><td colspan="3">低电压检测功能</td></tr>
<tr><td colspan="3">外部中断</td></tr>
</table>

**注**　**1.** 仅 48 引脚产品。
　　**2.** 仅 $\mu$PD78F0514, 78F0515, 和 78F0515D 。

**备注**　$f_{RH}$:　内部高速振荡时钟
　　$f_X$:　X1 时钟
　　$f_{EXCLK}$:　外部主系统时钟
　　$f_{XT}$:　XT1 时钟
　　$f_{EXCLKS}$:　外部副系统时钟
　　$f_{RL}$:　内部低速振荡时钟

### 表 20-1. HALT 模式中的操作状态 (2/2)

| 项目 \ HALT 模式设置 | | | 当 CPU 使用副系统时钟并执行 HALT 指令时 | |
|---|---|---|---|---|
| | | | 当 CPU 使用 XT1 时钟($f_{XT}$)时 | 当 CPU 使用外部副系统时钟($f_{EXCLKS}$)时 |
| 系统时钟 | | | 停止 CPU 时钟 | |
| | 主系统时钟 | $f_{RH}$ | 保持设置 HALT 模式前的状态 | |
| | | $f_X$ | | |
| | | $f_{EXCLK}$ | 由外部时钟输入进行操作或停止 | |
| | 副系统时钟 | $f_{XT}$ | 操作继续 (不能停止) | 保持设置 HALT 模式前的状态 |
| | | $f_{EXCLKS}$ | 由外部时钟输入进行操作或停止 | 操作继续 (不能停止) |
| | $f_{RL}$ | | 保持设置 HALT 模式前的状态 | |
| CPU | | | 操作停止 | |
| Flash 存储器 | | | | |
| RAM | | | 保持设置 HALT 模式前的状态 | |
| 端口(锁存器) | | | | |
| 16 位定时器/事件计数器 00[注 1] | | | 可操作 | |
| 8 位定时器/事件计数器 | 50[注 1] | | | |
| | 51[注 1] | | | |
| 8 位定时器 | H0 | | | |
| | H1 | | | |
| 钟表定时器 | | | | |
| 看门狗定时器 | | | 可操作。当通过可选字节设置“可以由软件停止内部低速振荡器”时，停止看门狗定时器的时钟。 | |
| 时钟输出[注 2] | | | 可操作 | |
| A/D 转换器 | | | 可操作，但是当外围硬件时钟 ($f_{PRS}$) 停止时，操作停止。 | |
| 串行接口 | UART0 | | 可操作 | |
| | UART6 | | | |
| | CSI10[注 1] | | | |
| | IIC0[注 1] | | | |
| 乘法器/除法器[注 3] | | | | |
| 上电清零（POC）功能 | | | | |
| 低电压检测功能 | | | | |
| 外部中断 | | | | |

注　**1.** 当 CPU 使用副系统时钟且内部高速振荡时钟已经停止时，不要启动使用外部时钟的外围硬件的功能。
**2.** 仅 48 引脚产品。
**3.** 仅μPD78F0514, 78F0515, 和 78F0515D 。

**备注**
$f_{RH}$:　内部高速振荡时钟
$f_X$:　X1 时钟
$f_{EXCLK}$:　外部主系统时钟
$f_{XT}$:　XT1 时钟
$f_{EXCLKS}$:　外部副系统时钟
$f_{RL}$:　内部低速振荡时钟

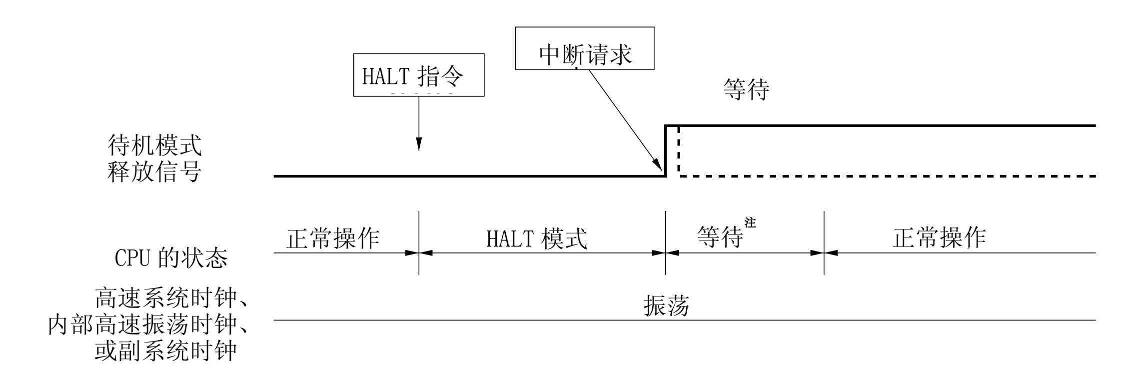
**(2) 释放 HALT 模式**

以下两种中断源可以释放 HALT 模式。

**(a) 由没有被屏蔽的中断请求释放**

当产生一个没有被屏蔽的中断时，释放 HALT 模式。如果允许响应中断，则执行向量中断服务程序。如果禁止响应中断，则执行下一个地址的指令。

**图 20-3. 通过产生中断请求释放 HALT 模式**




注　等待时间如下所示：

- 执行向量中断服务程序时: 8 或 9 个 时钟
- 不执行向量中断服务程序时: 2 或 3 个 时钟

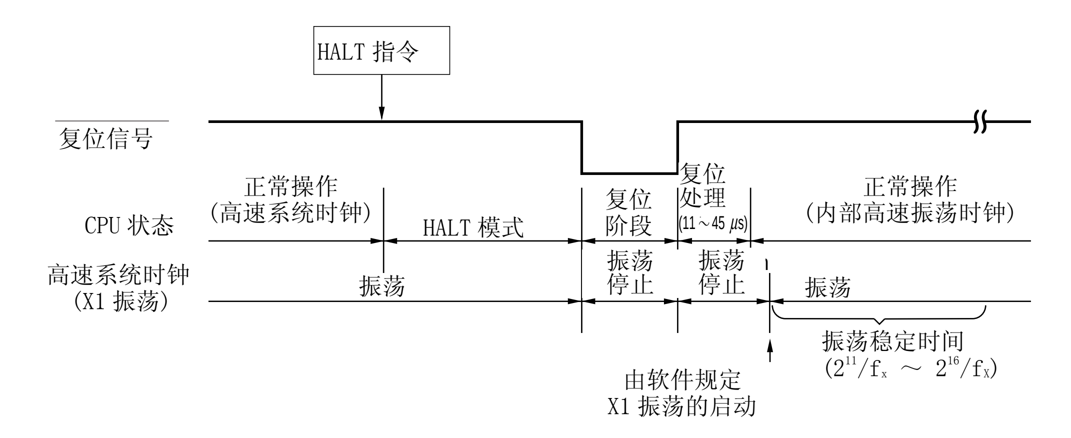
备注　　虚线表示释放待机模式的中断请求被响应的情况。

**(b) 通过产生的复位信号释放**

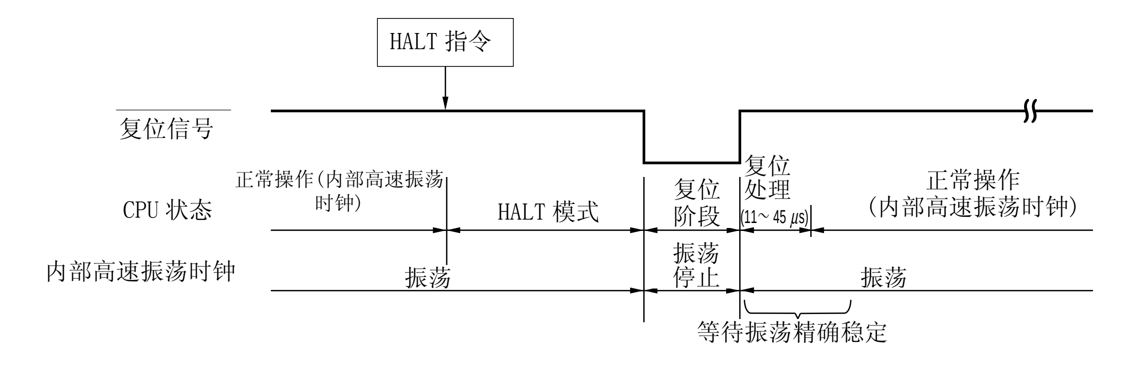
当产生复位信号时，释放 HALT 模式，然后在进行正常复位操作后，程序从复位向量指向的地址处开始执行。

<R> **图 20-4. 通过复位释放 HALT 模式**

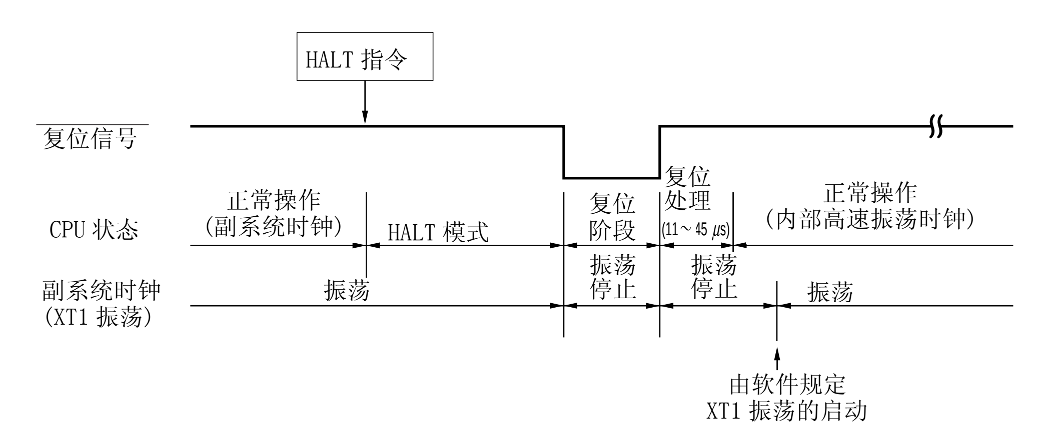
**(1) CPU 使用高速系统时钟**




**(2) CPU 使用内部高速振荡时钟**




**(3) CPU 使用副系统时钟**




**备注** fx: X1 时钟振荡频率

表 **20-2.** 在 **HALT** 模式中与中断请求对应的操作

| 释放源 | MK×× | PR×× | IE | ISP | 操作 |
|---|---|---|---|---|---|
| 可屏蔽中断请求 | 0 | 0 | 0 | × | 执行下一个地址的指令 |
| | 0 | 0 | 1 | × | 执行中断服务程序 |
| | 0 | 1 | 0 | 1 | 执行下一个地址的指令 |
| | 0 | 1 | × | 0 | |
| | 0 | 1 | 1 | 1 | 执行中断服务程序 |
| | 1 | × | × | × | 保持 HALT 模式 |
| 复位 | – | – | × | × | 复位处理 |

×：不必考虑

### 20.2.2 STOP 模式

**(1) STOP 模式设置及操作状态**

通过执行 STOP 指令设置 STOP 模式。仅当设置 STOP 模式前 CPU 使用主系统时钟时才可以设置 STOP 模式。

注意事项　**由于中断请求信号用于释放待机模式，因此如果一个中断源的中断请求标志被设置且中断屏蔽标志被清零时，则立即释放待机模式。因此，在执行 STOP 指令后，立即将 STOP 模式复位到 HALT 模式，并且在经历了 OSTS 设置的等待时间后，系统返回操作模式。**

STOP 模式中的操作状态如下所示。

### 表 20-3. STOP 模式中的操作状态

| STOP 模式设置 / 项目 | | | 当 CPU 使用主系统时钟并执行 STOP 指令时 | | |
|---|---|---|---|---|---|
| | | | 当 CPU 使用内部高速振荡时钟($f_{RH}$)时 | 当 CPU 使用 X1 时钟($f_X$)时 | 当 CPU 使用外部主系统时钟($f_{EXCLK}$)时 |
| 系统时钟 | | | 停止 CPU 时钟 | | |
| | 主系统时钟 | $f_{RH}$ | 停止 | | |
| | | $f_X$ | | | |
| | | $f_{EXCLK}$ | 输入无效 | | |
| | 副系统时钟 | $f_{XT}$ | 保持设置 STOP 模式前的状态 | | |
| | | $f_{EXCLKS}$ | 由外部时钟输入进行操作或停止 | | |
| | $f_{RL}$ | | 保持设置 STOP 模式前的状态 | | |
| CPU | | | 操作停止 | | |
| Flash 存储器 | | | | | |
| RAM | | | 保持设置 STOP 模式前的状态 | | |
| 端口 (锁存器) | | | | | |
| 16 位定时器/事件计数器 00[注 1] | | | 操作停止 | | |
| 8 位定时器/事件计数器 | 50[注 1] | | 仅当选择 TI50 作为计数时钟时可操作 | | |
| | 51[注 1] | | 仅当选择 TI51 作为计数时钟时可操作 | | |
| 8 位定时器 | H0 | | 在 8 位定时器/事件计数器 50 操作期间，仅当选择 TM50 输出作为计数时钟时可操作 | | |
| | H1 | | 仅当选择$f_{RL}$, $f_{RL}/2^7$, $f_{RL}/2^9$ 作为计数时钟时可操作 | | |
| 钟表定时器 | | | 仅当选择副系统时钟作为计数时钟时可操作 | | |
| 看门狗定时器 | | | 可操作。当可选字节设置“可由软件停止内部低速振荡器”时，停止看门狗定时器的时钟。 | | |
| 时钟输出[注 2] | | | 仅当选择副系统时钟作为计数时钟时可操作 | | |
| A/D 转换器 | | | 操作停止 | | |
| 串行接口 | UART0 | | 在 8 位定时器/事件计数器 50 操作期间，仅当选择 TM50 输出作为计数时钟时可操作 | | |
| | UART6 | | | | |
| | CSI10[注 1] | | 仅当选择外部时钟作为串行时钟时可操作 | | |
| | IIC0[注 1] | | 仅当选择从 EXSCL0/P62 引脚输入的外部时钟作为串行时钟时可操作 | | |
| 乘法器/除法器[注 3] | | | 操作停止 | | |
| 上电清零（POC）功能 | | | 可操作 | | |
| 低电压检测功能 | | | | | |
| 外部中断 | | | | | |

<R>

<R> **注** **1.** 要在 STOP 模式下，使用从外围硬件引脚输入的外部时钟时，不要打开这些功能。

**2.** 仅 48 引脚产品。

**3.** 仅$\mu$PD78F0514, 78F0515, 和 78F0515D 。

**备注** $f_{RH}$: 内部高速振荡时钟

$f_X$: X1 时钟

$f_{EXCLK}$: 外部主系统时钟

$f_{XT}$: XT1 时钟

$f_{EXCLKS}$: 外部副系统时钟

$f_{RL}$: 内部低速振荡时钟

注意事项 1. 在 **STOP** 模式释放后，要使用在 **STOP** 模式下停止操作的外围硬件，以及在 **STOP** 模式下时钟停止操作的外围硬件，必须重新启动外围硬件。

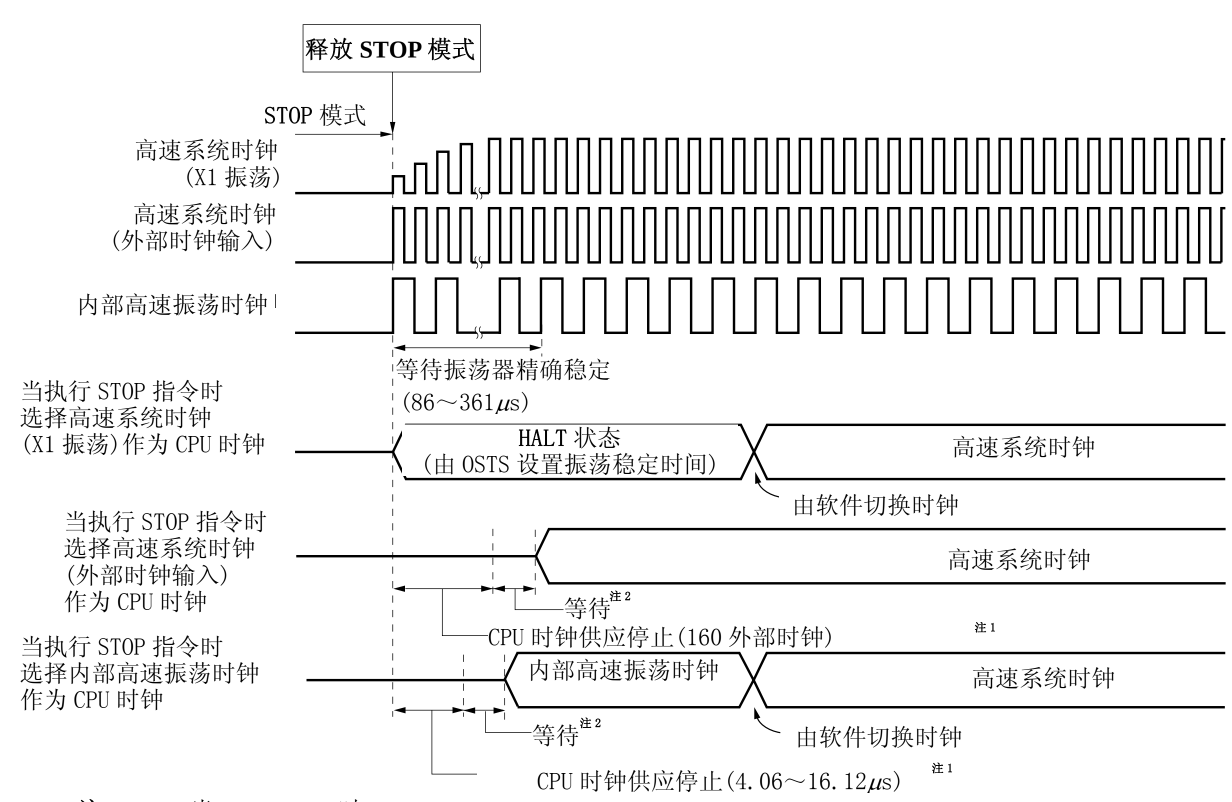
2. 即使通过可选字节选择“可由软件停止内部低速振荡器”，在 **STOP** 模式下，内部低速振荡时钟仍继续振荡，并保持 **STOP** 模式设置前的状态。要在 **STOP** 模式下停止内部低速振荡器的振荡，应通过软件停止该振荡器，然后执行 **STOP** 指令。

3. 当 **CPU** 使用高速系统时钟**(X1** 振荡**)**时，为了在 **STOP** 模式释放后缩短振荡稳定时间，可在执行 **STOP** 指令前将 **CPU** 时钟暂时切换到内部高速振荡时钟。**STOP** 模式释放后，在将 **CPU** 时钟从高速振荡时钟切换到高速系统时钟 **(X1** 振荡**)** 之前，应使用振荡稳定时间计数器的状态寄存器**(OSTC)**检测振荡稳定时间。

<R> 4. 当 **AMPH = 1** 时执行 **STOP** 指令，当内部高速振荡时钟作为 **CPU** 时钟时，在 **STOP** 模式释放后 **CPU** 时钟停止 **4.06** ～ **16.12** $\mu$**s**，或者当高速系统时钟（外部时钟输入）作为 **CPU** 时钟时，**CPU** 时钟停止 **160** 个外部时钟周期的时间。

**(2)** 释放 **STOP** 模式

<R> 图 **20-5.** 释放 **STOP** 模式时的操作时序 **(**当产生没有屏蔽的中断请求时**)**




注 **1.** 当 AMPH = 1 时。

**2.** 等待时间如下所示：

- 执行向量中断服务程序时: 8 或 9 个 时钟
- 不执行向量中断服务程序时: 2 或 3 个 时钟

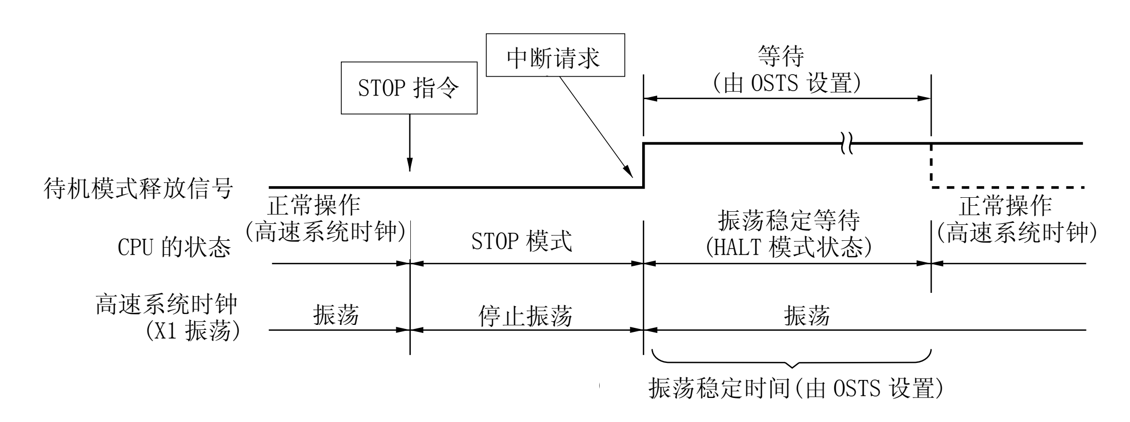
STOP 模式可通过以下两种方式释放。

**(a)** 由没有被屏蔽的中断请求释放

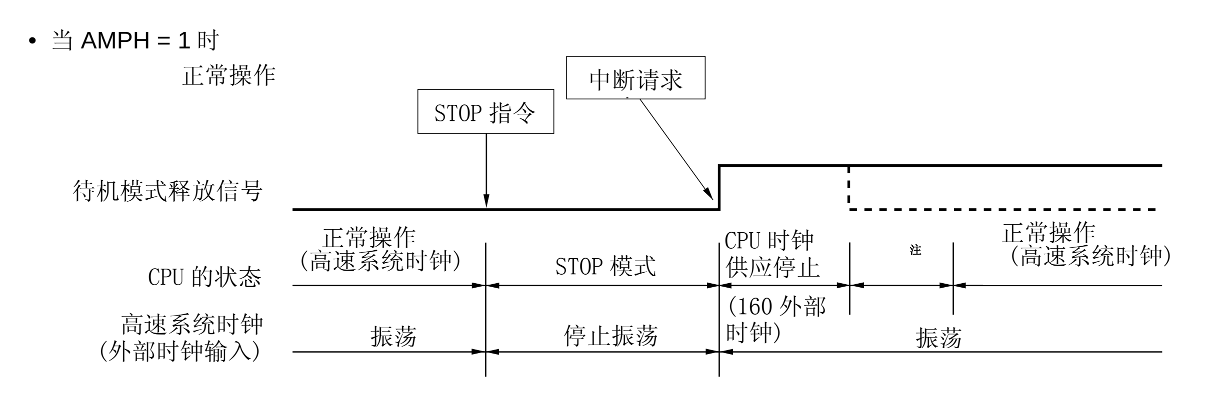
当产生一个没有被屏蔽的中断请求时，释放 STOP 模式。经历振荡稳定时间后，如果允许响应中断，则执行向量中断服务程序。如果禁止响应中断，则执行下一个地址的指令。

<R> **图 20-6. 通过产生中断请求释放 STOP 模式 (1/2)**

**(1) 当 CPU 使用高速系统时钟时 (X1 振荡器)**




**(2) 当 CPU 使用高速系统时钟时 (外部时钟输入) (1/2)**




**注** 等待时间如下所示：

- 执行向量中断服务程序时: 8 或 9 个 时钟
- 不执行向量中断服务程序时: 2 或 3 个 时钟周期

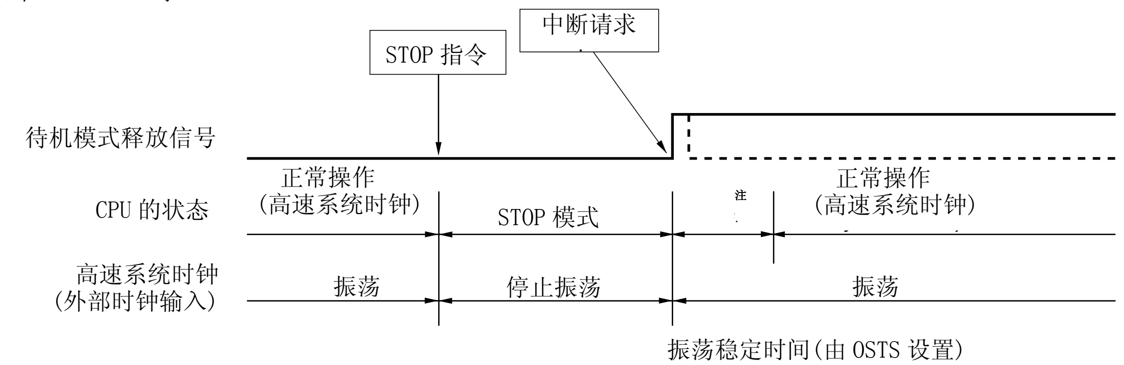
**备注** 虚线表示释放待机模式的中断请求被响应的情况。

<R>

**图 20-6. 通过产生中断请求释放 STOP 模式 (2/2)**

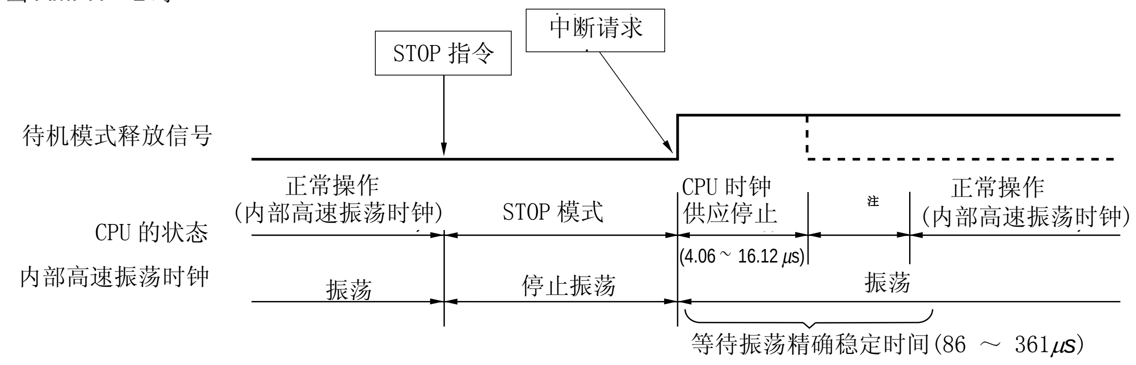
**(2) 当 CPU 使用高速系统时钟时 (外部时钟输入) (2/2)**

• 当 AMPH = 0 时

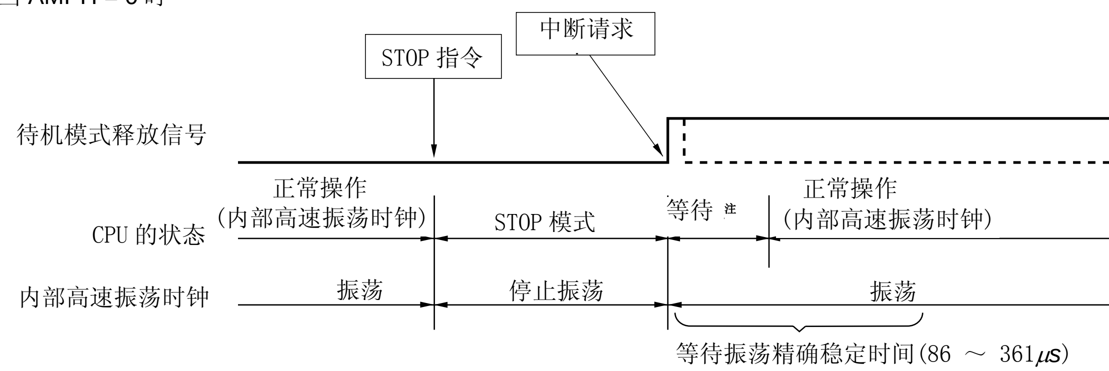



**(3) 当 CPU 使用内部高速振荡时钟时**

• 当 AMPH = 1 时




• 当 AMPH = 0 时




**注**　等待时间如下所示：

- 执行向量中断服务程序时: 8 或 9 个 时钟
- 不执行向量中断服务程序时: 2 或 3 个 时钟

**备注**　　虚线表示释放待机模式的中断请求被响应的情况。

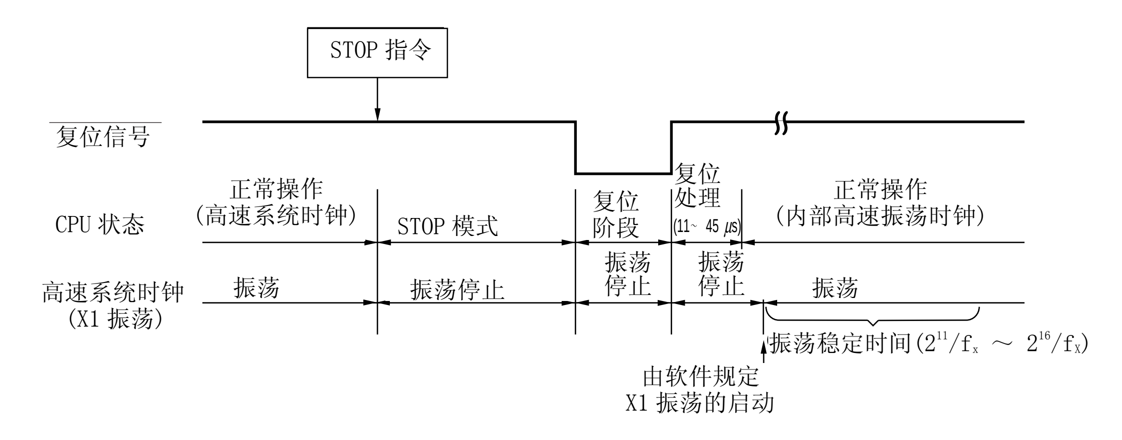
**(b)** 由复位信号的产生释放

当产生复位信号时，释放 STOP 模式，然后在正常复位操作情况下，当程序转向复位向量地址后执行之。

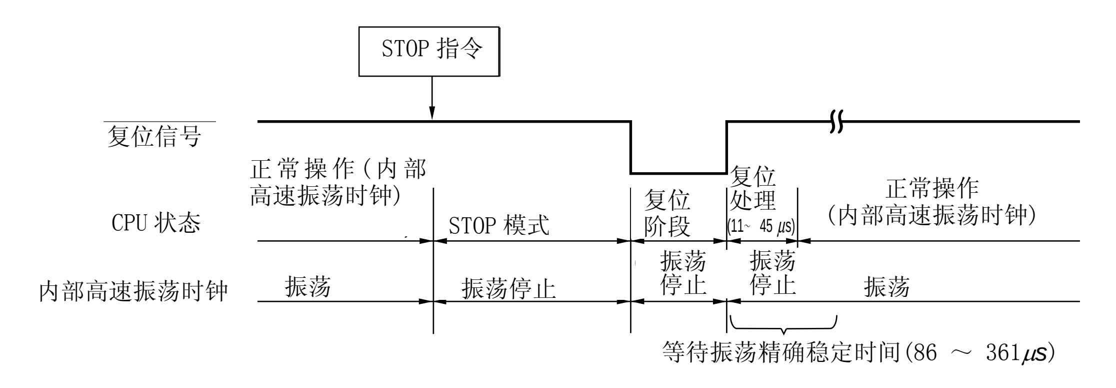
<R>

**图 20-7. 由复位释放 STOP 模式**

**(1) 当 CPU 使用高速系统时钟**




**(2) 当 CPU 使用内部高速振荡时钟**




**备注**　fx: X1 时钟振荡频率。

**表 20-4. 在 STOP 模式中与中断请求对应的操作**

| 释放源 | MK×× | PR×× | IE | ISP | 操作 |
|---|---|---|---|---|---|
| 可屏蔽的中断请求 | 0 | 0 | 0 | × | 执行下一个地址的指令 |
| | 0 | 0 | 1 | × | 执行中断服务程序 |
| | 0 | 1 | 0 | 1 | 执行下一个地址的指令 |
| | 0 | 1 | × | 0 | |
| | 0 | 1 | 1 | 1 | 执行中断服务程序 |
| | 1 | × | × | × | 保持 STOP 模式 |
| 复位 | – | – | × | × | 复位处理 |

×: 不必考虑

# 第二十一章　复位功能

以下 4 种操作用于产生复位信号。

(1)　由 $\overline{\text{RESET}}$ 引脚输入的外部复位信号
(2)　由看门狗定时器程序循环检测引起的内部复位
(3)　通过比较上电清零（POC）电路的检测电压和供电电压引起的内部复位
(4)　通过比较低电压检测电路 （LVI）的检测电压和供电电压引起的内部复位

外部复位与内部复位在功能上没有什么区别。在这两种情况中，当产生复位信号时，程序都是从地址 0000H 和 0001H 处开始执行。

当 $\overline{\text{RESET}}$ 引脚输入为低电平、看门狗定时器溢出或 POC 和 LVI 电路的电压检测都可以引起复位，每项硬件的设置状态如表 21-1 和 21-2 所示。在复位信号产生期间或在复位释放后振荡稳定时间内，除 P130（低电平输出）以外，每个引脚均为高阻抗。

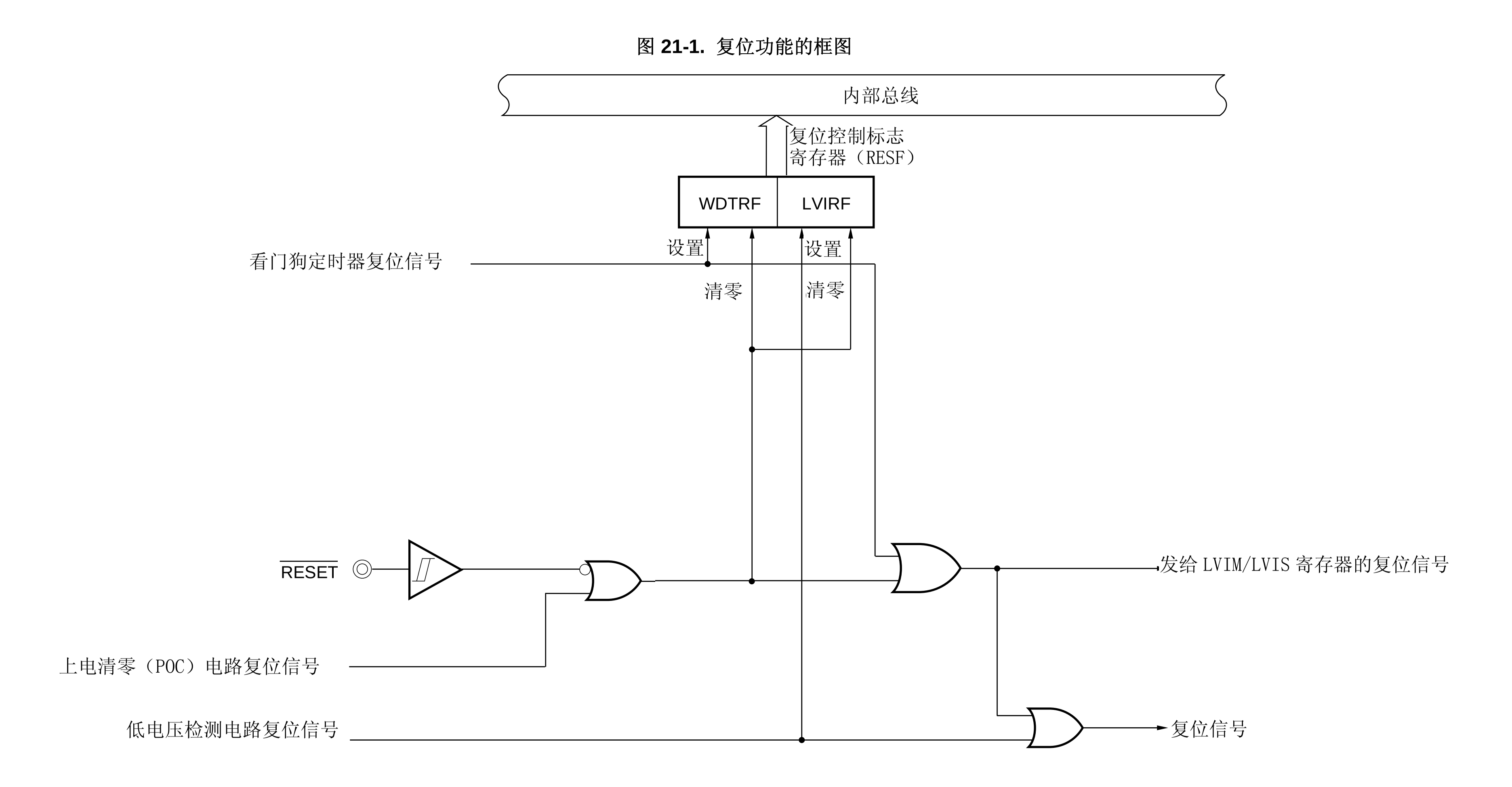
当 $\overline{\text{RESET}}$ 引脚输入低电平时，设备被复位。当 $\overline{\text{RESET}}$ 引脚输入高电平时，设备从复位状态释放，并在复位处理后使用内部高速振荡时钟执行程序。由看门狗定时器引起的复位自动释放，并在复位处理后使用内部高速振荡时钟执行程序(参见**图 21-2 ～ 21-4**)。由 POC 和 LVI 电路供电电压检测引起的复位，在复位后，当 $V_{DD} \geq V_{POC}$ 或 $V_{DD} \geq V_{LVI}$ 时自动释放，并使用内部高速振荡时钟执行程序（参见**第二十二章 上电清零电路** 和 **第二十三章 低电压检测电路**）。

注意事项　**1.** 对于外部复位，输入 $\overline{\text{RESET}}$ 引脚的低电平的时间至少应为 **10 μs**。

**2.** 复位输入期间，**X1** 时钟、**XT1** 时钟、内部高速振荡时钟和内部低速振荡时钟停止振荡。外部主系统时钟输入和外部子系统时钟输入无效。

**3.** 在通过复位释放 **STOP** 模式时，复位输入期间保持 **STOP** 模式的内容。但 **P130**（低电平输出）除外的端口引脚变为高阻态。

**图 21-1. 复位功能的框图**

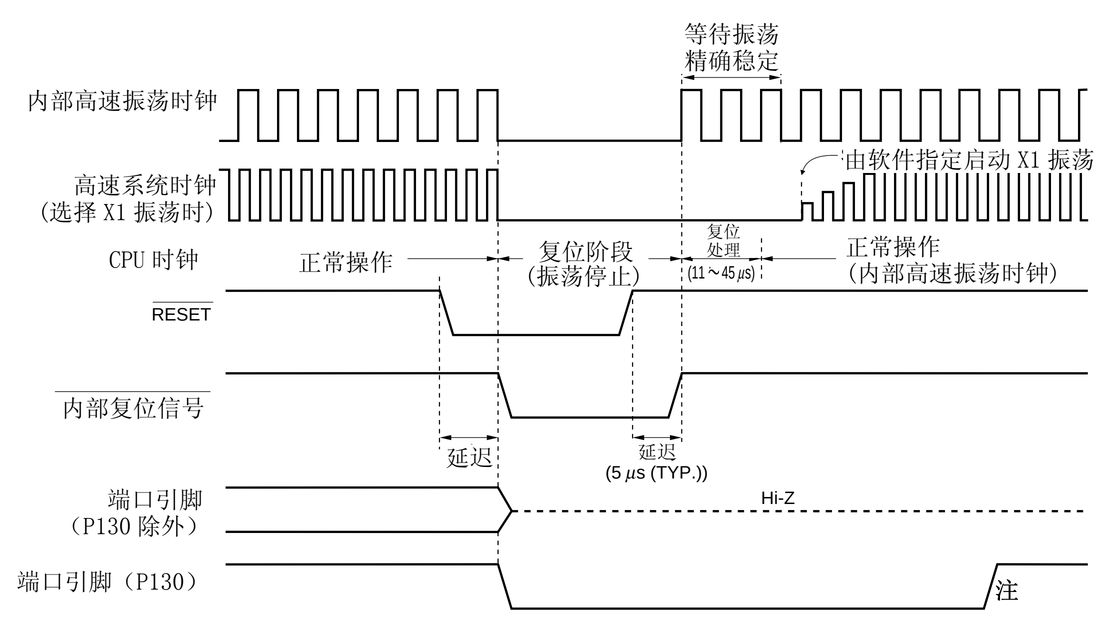



**注意事项　LVI 电路内部复位功能不能复位 LVI 电路。**

**备注**　**1.** LVIM: 低电压检测寄存器
　　　　**2.** LVIS: 低电压检测等级选择寄存器

<R>

**图 21-2. 由 RESET 输入进行复位的时序**



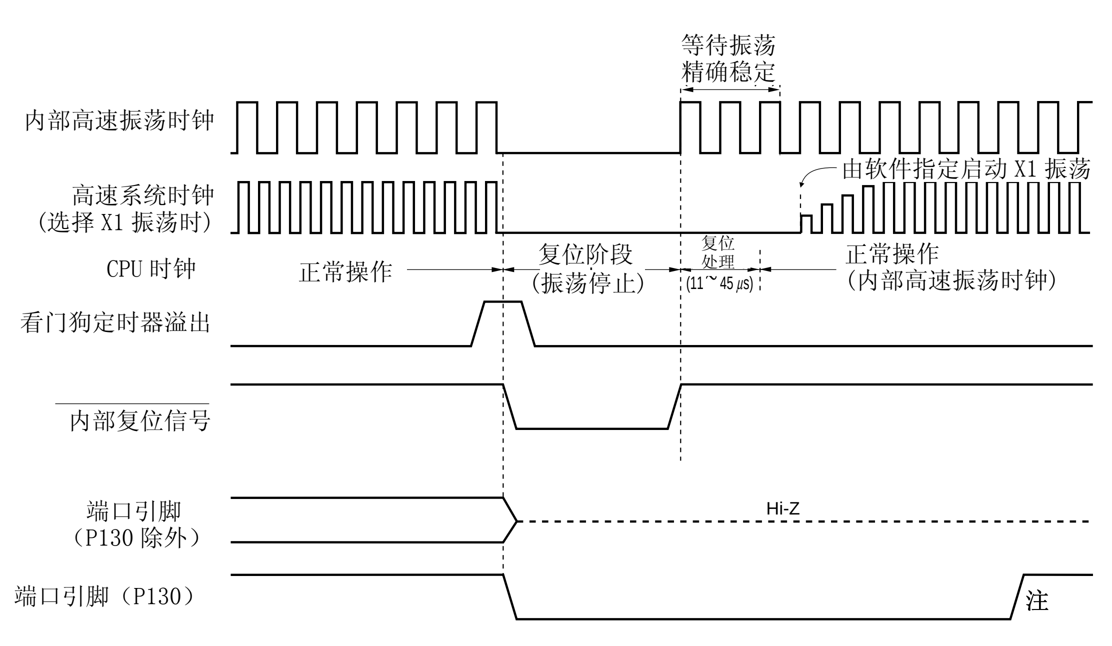

**备注**　当复位有效时，P130 [注1]的输出为低电平。如果在复位有效前将P130[注1]设置为高电平输出，则P130[注1]的输出信号可以作为虚拟的CPU复位信号。

注　**1.** 仅 48 引脚产品。

**2.** 由软件将 P130 设置为高电平输出。

<R>

**图 21-3. 由于看门狗定时器溢出进行复位的时序**




**备注**　当复位有效时，P130 [注1]的输出为低电平。如果在复位有效前将P130[注1]设置为高电平输出，则P130[注1]的输出信号可以作为虚拟的CPU复位信号。

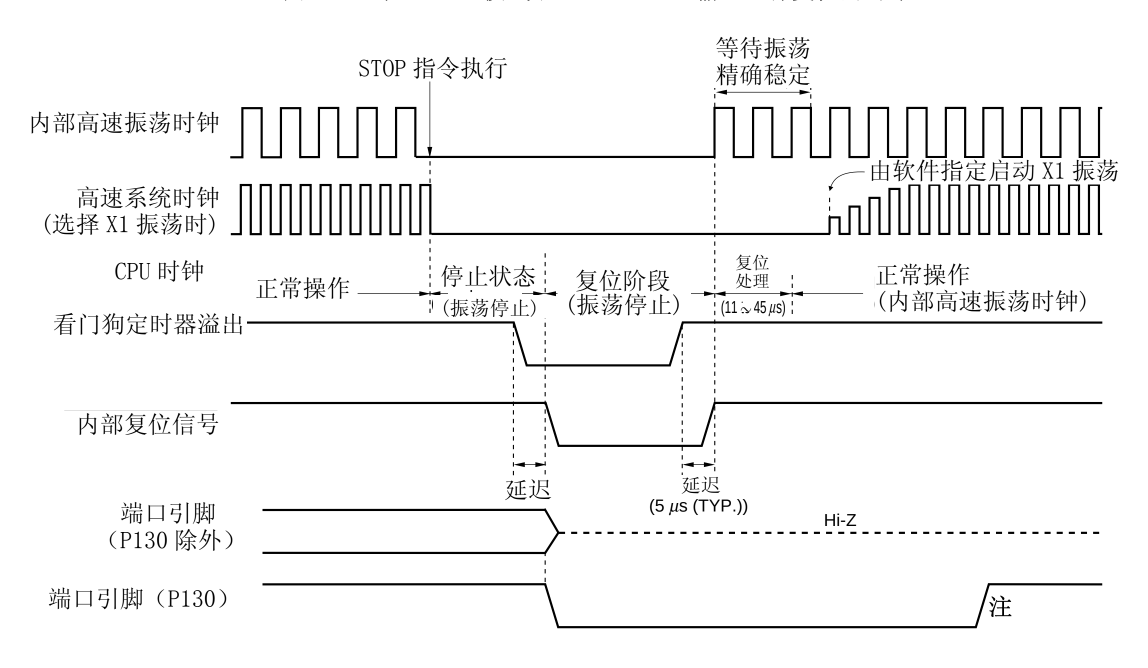
**注意事项　看门狗定时器内部复位功能也可以复位看门狗定时器。**

注　**1.** 仅 48 引脚产品。

**2.** 由软件将 P130 设置为高电平输出。

<R>

图 21-4. 在 STOP 模式下通过 $\overline{\text{RESET}}$ 输入进行复位的时序




备注　**1.** 当复位有效时，P130 注 1的输出为低电平。如果在复位有效前将P130注 1设置为高电平输出，则P130注 1的输出信号可以作为虚拟的CPU复位信号。

**2.** 如需了解上电清零（POC）电路和低电压检测电路的复位时序，可参见 **第二十二章　上电清零电路** 和 **第二十三章　低电压检测电路**。

注　**1.** 仅 48 引脚产品。

**2.** 由软件将 P130 设置为高电平输出。

表 **21-1.** 复位期间的操作状态

| 项目 | | | 复位期间 |
|---|---|---|---|
| 系统时钟 | | | 停止 CPU 时钟 |
| | 主系统时钟 | $f_{RH}$ | 停止操作 |
| | | $f_X$ | 停止操作(引脚为 I/O 端口模式) |
| | | $f_{EXCLK}$ | 时钟输入无效 (引脚为 I/O 端口模式) |
| | 副系统时钟 | $f_{XT}$ | 停止操作 (引脚为 I/O 端口模式) |
| | | $f_{EXCLKS}$ | 时钟输入无效 (引脚为 I/O 端口模式) |
| | $f_{RL}$ | | 停止操作 |
| CPU | | | |
| flash 存储器 | | | |
| RAM | | | |
| 端口 (锁存器) | | | |
| 16 位定时器/事件计数器 00 | | | |
| 8 位定时器/事件计数器 | 50 | | |
| | 51 | | |
| 8 位定时器 | H0 | | |
| | H1 | | |
| 钟表定时器 | | | |
| 看门狗定时器 | | | |
| 时钟输出[注 1] | | | |
| A/D 转换器 | | | |
| 串行接口 | UART0 | | |
| | UART6 | | |
| | CSI10 | | |
| | IIC0 | | |
| 乘法器/除法器[注 2] | | | |
| 上电清零功能 | | | 可操作 |
| 低电压检测功能 | | | 停止操作 |
| 外部中断 | | | |

注　**1.** 仅 48 引脚产品。

　　**2.** 仅μPD78F0514, 78F0515, 和 78F0515D 。

备注　$f_{RH}$:　内部高速振荡时钟

　　$f_X$:　X1 振荡时钟

　　$f_{EXCLK}$:　外部主系统时钟

　　$f_{XT}$:　XT1 振荡时钟

　　$f_{EXCLKS}$:　外部子系统时钟

　　$f_{RL}$:　内部低速振荡时钟

表 21-2. 复位响应后的硬件状态 (1/3)

| 硬件 | | 响应复位后的状态[注1] |
|---|---|---|
| 程序计数器 （PC） | | 设置复位向量表的内容（0000H、0001H） |
| 堆栈指针 （SP） | | 不确定 |
| 程序状态字(PSW) | | 02H |
| RAM | 数据存储器 | 不确定[注2] |
| | 通用寄存器 | 不确定[注2] |
| 端口寄存器 （P0 ～ P4, P6, P7, P12 , P13[注3], P14[注3]） （输出锁存器） | | 00H |
| 端口模式寄存器 (PM0 ～ PM4, PM6, PM7, PM12, PM14[注3]) | | FFH |
| 上拉电阻选择寄存器 (PU0, PU1, PU3, PU4, PU7, PU12, PU14[注3]) | | 00H |
| 内部扩展 RAM 容量切换寄存器(IXS) | | 0CH[注4] |
| 内存容量切换寄存器(IMS) | | CFH[注4] |
| 时钟操作模式选择寄存器 (OSCCTL) | | 00H |
| 处理器时钟控制寄存器(PCC) | | 01H |
| 内部振荡模式寄存器(RCM) | | 80H |
| 主 OSC 控制寄存器(MOC) | | 80H |
| 主时钟模式寄存器(MCM) | | 00H |
| 振荡稳定时间计数器的状态寄存器(OSTC) | | 00H |
| 振荡稳定时间选择寄存器(OSTS) | | 05H |
| 16 位定时器/事件计数器 00 | 定时器计数器 00 (TM00) | 0000H |
| | 捕捉/比较寄存器 000, 010 (CR000, CR010) | 0000H |
| | 模式控制寄存器 00 (TMC00) | 00H |
| | 预分频器模式寄存器 00 (PRM00) | 00H |
| | 捕捉/比较控制寄存器 00 (CRC00) | 00H |
| | 定时器输出控制寄存器 00 (TOC00) | 00H |

注　**1.** 在复位信号产生或振荡稳定时间等待期间，硬件状态中只有 PC 内容不确定。复位后其它硬件状态保持不变。

**2.** 当在待机模式中进行复位时，即使复位后也保持复位前的状态。

**3.** 仅 48 引脚产品。

**4.** 在所有 78K0/KC2 产品中，无论内存容量如何，当复位释放后内存容量切换寄存器(IMS)和内部扩展 RAM 容量切换寄存器(IXS)的初值为常数(IMS = CFH, IXS = 0CH)。因此复位释放后必须为每个产品设置以下值。

| Flash 存储器版本 (78K0/KC2) | IMS | IXS |
|---|---|---|
| μPD78F0511 | 04H | 0CH |
| μPD78F0512 | C6H | |
| μPD78F0513, 78F0513D | C8H | |
| μPD78F0514 | CCH | 0AH |
| μPD78F0515, 78F0515D | CFH | 08H |

表 21-2. 复位响应后的硬件状态 (2/3)

| 硬件 | | 响应复位后的状态[注 1] |
|---|---|---|
| 8 位定时器/事件计数器 50, 51 | 定时器计数器 50, 51 (TM50, TM51) | 00H |
| | 比较寄存器 50, 51 (CR50, CR51) | 00H |
| | 定时器时钟选择寄存器 50, 51 (TCL50, TCL51) | 00H |
| | 模式控制寄存器 50, 51 (TMC50, TMC51) | 00H |
| 8 位定时器 H0, H1 | 比较寄存器 00, 10, 01, 11 (CMP00, CMP10, CMP01, CMP11) | 00H |
| | 模式寄存器 (TMHMD0, TMHMD1) | 00H |
| | 载波控制寄存器 1 (TMCYC1)[注 2] | 00H |
| 钟表定时器 | 操作模式寄存器(WTM) | 00H |
| 时钟输出控制器[注 3] | 时钟输出选择寄存器 (CKS) | 00H |
| 看门狗定时器 | 使能寄存器 (WDTE) | 1AH/9AH[注 4] |
| A/D 转换器 | 10 位 A/D 转换结果寄存器 (ADCR) | 0000H |
| | 8 位 A/D 转换结果寄存器(ADCRH) | 00H |
| | 模式寄存器 (ADM) | 00H |
| | 模拟输入通道选择寄存器(ADS) | 00H |
| | A/D 端口配置寄存器 (ADPC) | 00H |
| 串行接口 UART0 | 接收缓冲寄存器 0 (RXB0) | FFH |
| | 发送移位寄存器 0 (TXS0) | FFH |
| | 异步串行接口操作模式寄存器 0 (ASIM0) | 01H |
| | 异步串行接口接收错误状态寄存器 0 (ASIS0) | 00H |
| | 波特率发生器控制寄存器 0 (BRGC0) | 1FH |
| 串行接口 UART6 | 接收缓冲寄存器 6 (RXB6) | FFH |
| | 发送缓冲寄存器 6 (TXB6) | FFH |
| | 异步串行接口操作模式寄存器 6 (ASIM6) | 01H |
| | 异步串行接口接收错误状态寄存器 6 (ASIS6) | 00H |
| | 异步串行接口发送状态寄存器 6 (ASIF6) | 00H |
| | 时钟选择寄存器 6 (CKSR6) | 00H |
| | 波特率发生器控制寄存器 6 (BRGC6) | FFH |
| | 异步串行接口控制寄存器 6 (ASICL6) | 16H |
| | 输入切换控制寄存器 (ISC) | 00H |
| 串行接口 CSI10 | 发送缓冲器寄存器 10 (SOTB10) | 00H |
| | 串行 I/O 移位寄存器 10 (SIO10) | 00H |
| | 串行操作模式寄存器 10 (CSIM10) | 00H |
| | 串行时钟选择寄存器 10 (CSIC10) | 00H |

注　**1.** 在复位信号产生或振荡稳定时间等待期间，硬件状态中只有 PC 内容不确定。复位后其它硬件状态保持不变。

**2.** 仅用于 8 位定时器 H1。

**3.** 仅 48 引脚产品。

**4.** 通过设置选项字节确定 WDTE 的复位值。

表 21-2. 复位响应后的硬件状态 (3/3)

| 硬件 | | 响应复位后的状态[注 1] |
|---|---|---|
| 串行接口 IIC0 | 移位寄存器 0 (IIC0) | 00H |
| | 控制寄存器 0 (IICC0) | 00H |
| | 从设备地址寄存器 0 (SVA0) | 00H |
| | 时钟选择寄存器 0 (IICCL0) | 00H |
| | 功能扩展寄存器 0 (IICX0) | 00H |
| | 状态寄存器 0 (IICS0) | 00H |
| | 标志寄存器 0 (IICF0) | 00H |
| 乘法器/除法器[注 2] | 余数寄存器 0 (SDR0) | 0000H |
| | 乘/除数寄存器 A0 (MDA0H, MDA0L) | 0000H |
| | 乘/除数寄存器 B0 (MDB0) | 0000H |
| | 乘法器/除法器控制寄存器 0 (DMUC0) | 00H |
| 按键中断 | 按键返回模式寄存器(KRM) | 00H |
| 复位功能 | 复位控制标志寄存器 (RESF) | 00H[注 3] |
| 低电压检测电路 | 低电压检测寄存器(LVIM) | 00H[注 3] |
| | 低电压检测等级选择寄存器 (LVIS) | 00H[注 3] |
| 中断 | 请求标志寄存器 0L, 0H, 1L, 1H (IF0L, IF0H, IF1L, IF1H) | 00H |
| | 屏蔽标志寄存器 0L, 0H, 1L, 1H (MK0L, MK0H, MK1L, MK1H) | FFH |
| | 优先级指定标志寄存器 0L, 0H, 1L, 1H (PR0L, PR0H, PR1L, PR1H) | FFH |
| | 外部中断上升沿使能寄存器(EGP) | 00H |
| | 外部中断下降沿使能寄存器(EGN) | 00H |

注　**1.** 在复位信号产生或振荡稳定时间等待期间，硬件状态中只有 PC 内容不确定。复位后其它硬件状态保持不变。

**2.** 仅 μPD78F0514, 78F0515, 和 78F0515D 提供乘法器/除法器。

**3.** 这些值依据复位源的不同而变化。

| 寄存器 \ 复位源 | | $\overline{\text{RESET}}$ 输入 | 由 POC 进行复位 | 由 WDT 进行复位 | 由 LVI 进行复位 |
|---|---|---|---|---|---|
| RESF | WDTRF 位 | 清零(0) | 清零(0) | 设置 (1) | 保持 |
| | LVIRF 位 | | | 保持 | 设置 (1) |
| LVIM | | 清零(00H) | 清零 (00H) | 清零(00H) | 保持 |
| LVIS | | | | | |

## 21.1 确认复位源的寄存器

78K0/KC2 中有许多内部复位源。复位控制标志寄存器（RESF）用于存储产生复位请求的复位源。
可由 8 位存储器操作指令读取 RESF。
$\overline{\text{RESET}}$ 输入、由 POC 电路引起的复位输入以及读取 RESF 可将 RESF 清零（00H）。

**图 21-5. 复位控制标志寄存器（RESF）的格式**

地址: FFACH　复位后: 00H[注]　R

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| RESF | 0 | 0 | 0 | WDTRF | 0 | 0 | 0 | LVIRF |

| WDTRF | 看门狗定时器产生的内部复位请求(WDT) |
|---|---|
| 0 | 不产生内部复位请求，或对 RESF 清零 |
| 1 | 产生内部复位请求 |

| LVIRF | 由低电压检测电路（LVI）产生的内部复位请求 |
|---|---|
| 0 | 不产生内部复位请求，或对 RESF 清零 |
| 1 | 产生内部复位请求 |

**注**　复位后的值依据复位源而变化。

**注意事项　不能通过 1 位存储器操作指令读取数据。**

产生复位请求时的 RESF 状态如表 21-3 所示。

**表 21-3. 产生复位请求时的 RESF 状态**

| 标志 \ 复位源 | $\overline{\text{RESET}}$ 输入 | 由 POC 引起的复位 | 由 WDT 引起的复位 | 由 LVI 引起的复位 |
|---|---|---|---|---|
| WDTRF | 清零(0) | 清零(0) | 设置(1) | 保持 |
| LVIRF | | | 保持 | 设置 (1) |

# 第二十二章　上电清零电路

## 22.1 上电清零电路的功能

上电清零电路（POC）有以下功能。

- 在上电时产生内部复位信号。
  在 1.59 V POC 模式下 (可选字节: POCMODE = 0)，当供电电压 ($V_{DD}$) 超过 1.59 V ±0.15 V 时，释放复位信号。
  在 2.7 V/1.59 V POC 模式下 (可选字节: POCMODE = 1)，当供电电压 ($V_{DD}$) 超过 2.7 V ±0.2 V 时，释放复位信号。

<R>
- 比较供电电压（$V_{DD}$）和检测电压（$V_{POC}$ = 1.59 V ±0.15 V），并在 $V_{DD} < V_{POC}$ 时产生内部复位信号。

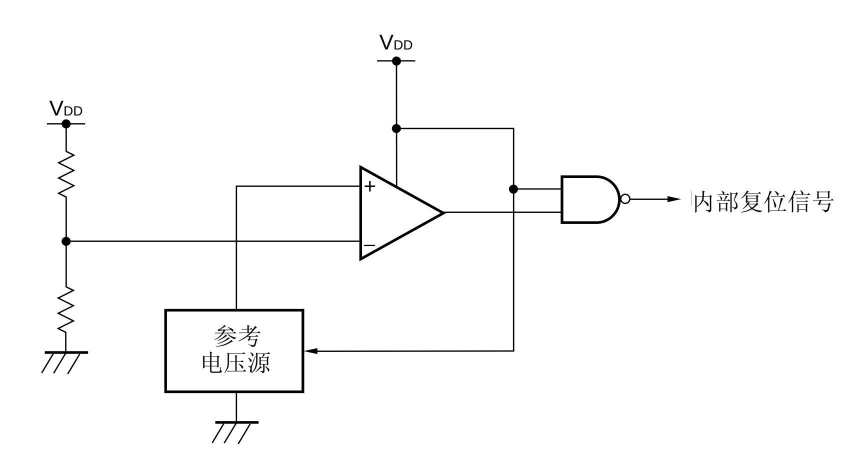
**注意事项　如果由 POC 电路产生内部复位信号，则将复位控制标志寄存器（RESF）清零（00H）。**

**备注**　该产品集成了多种可以产生内部复位信号的硬件功能。RESF 中有一个标志用于指示复位源，用于看门狗定时器（WDT）或低电压检测电路（LVI）产生的内部复位信号。当 WDT 或 LVI 产生内部复位信号时，不对 RESF 清零（00H），标志位被设置为 1。
如需了解 RESF 的详细内容，参见**第二十一章　复位功能**。

## 22.2 上电清零电路的配置

上电清零电路的框图如图 22-1 所示。

**图 22-1.** 上电清零电路的框图




## 23.3 上电清零电路的操作

**(1) 1.59 V POC 模式 (选项字节: POCMODE = 0)**

- 上电时产生内部复位信号。当供电电压($V_{DD}$)超过检测电压($V_{POC}$ = 1.59 V ±0.15 V)时，释放复位状态。
- 比较供电电压($V_{DD}$)和检测电压($V_{POC}$ = 1.59 V ±0.15 V)。当 $V_{DD} < V_{POC}$ 时，产生内部复位信号。当 $V_{DD} \geq V_{POC}$，释放内部复位信号。

**(2) 2.7 V/1.59 V POC 模式 (选项字节: POCMODE = 1)**

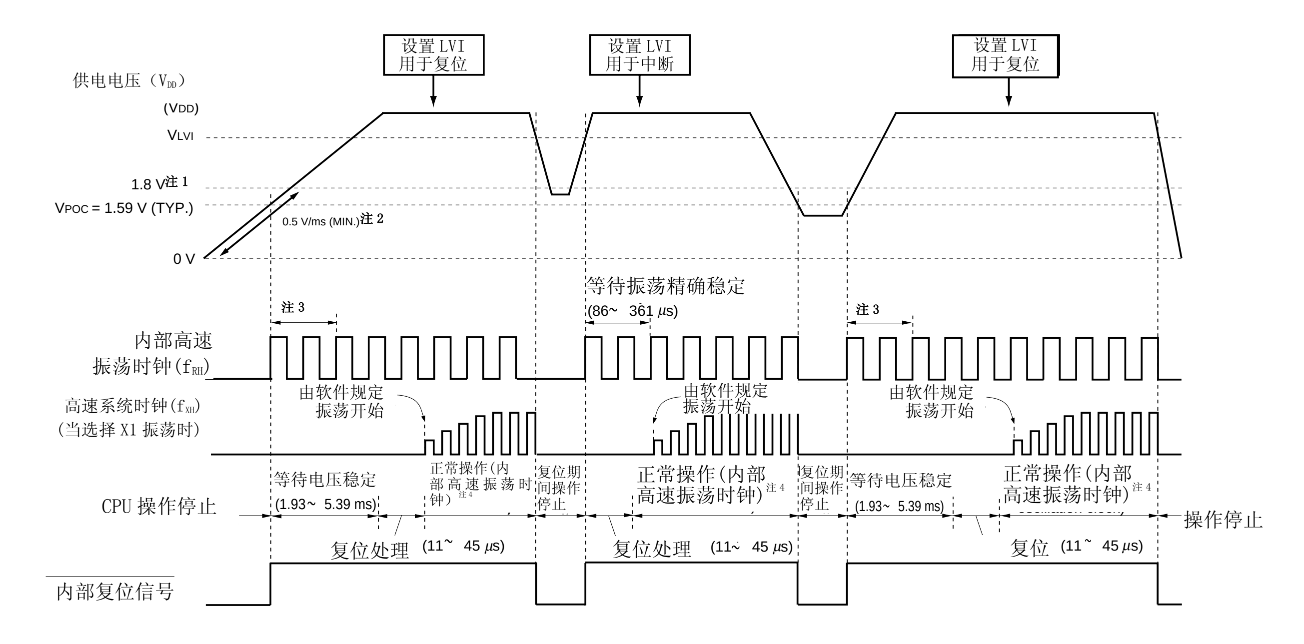
- 上电时产生内部复位信号。当供电电压($V_{DD}$)超过检测电压($V_{DDPOC}$ = 2.7 V ±0.2 V)时，释放复位状态。

<R>
- 比较供电电压($V_{DD}$)和检测电压($V_{POC}$ = 1.59 V ±0.15 V)。当 $V_{DD} < V_{POC}$ 时，产生内部复位信号。当 $V_{DD} \geq V_{DDPOC}$ 时，释放内部复位信号。

由上电清零电路和低电压检测电路产生内部复位信号的时序如下所示。

<R> **图 22-2. 由上电清零电路和低电压检测电路产生内部复位信号的时序 (1/2)**

**(1) 1.59 V POC 模式 (选项字节: POCMODE = 0)**




注 **1.** 保证操作的电压范围：1.8 V ≤ $V_{DD}$ ≤ 5.5 V。当供电电压下降时，为使状态低于 1.8 V 复位状态，可以使用低电压检测电路的复位功能，或通过 $\overline{\text{RESET}}$ 引脚输入低电平。

<R> **2.** 如果通电后电压上升到 1.8V 的速率小于 0.5 V/ms (MIN.)，则在上电后、电压抵达 1.8V 以前输入一个低电平到 $\overline{\text{RESET}}$ 引脚，或者通过使用可选字节 (POCMODE = 1) 设定 2.7 V/1.59 V POC 模式。

<R> **3.** 内部电压稳定时间包括内部高速振荡时钟的振荡精确稳定时间。

**4.** 可以选择内部高速振荡时钟和一个高速系统时钟或一个副系统时钟作为 CPU 时钟。如果要使用 X1 时钟，可以通过 OSTC 确认是否经历了振荡稳定时间。而如果要使用 XT1 时钟，可以通过定时器功能来确认是否经历了振荡稳定时间。

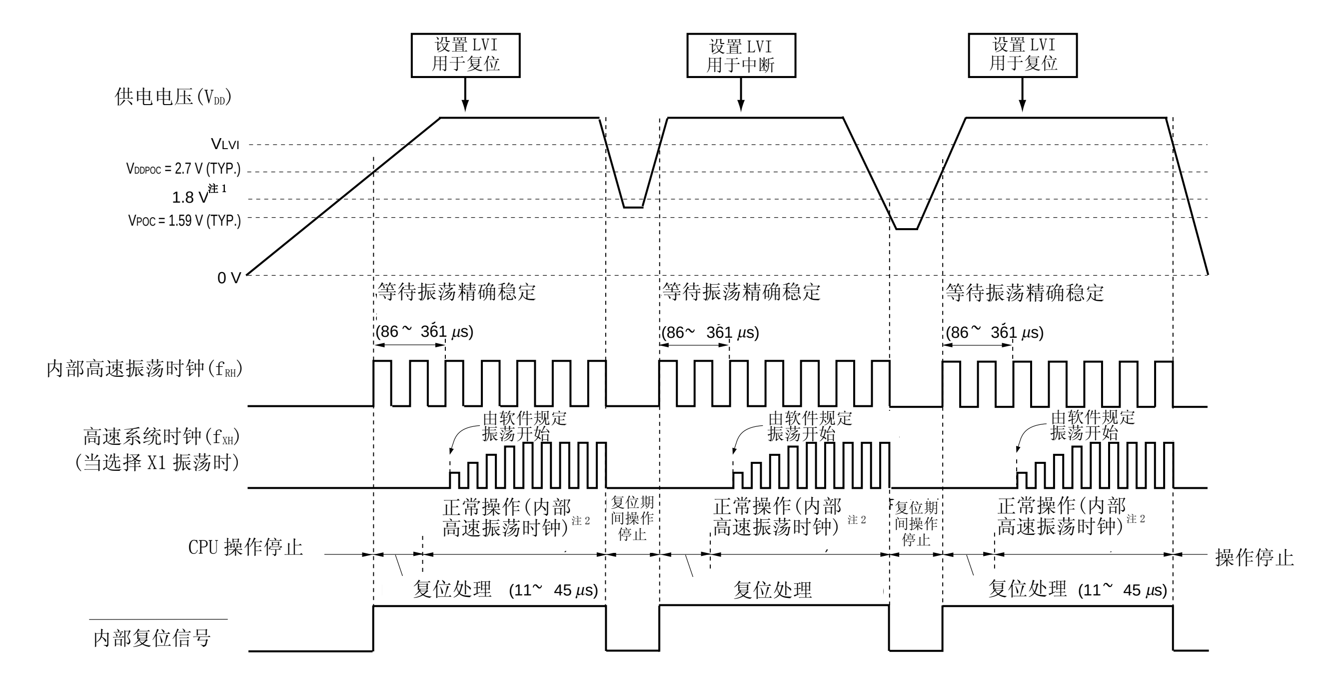
**注意事项　在释放复位状态后通过软件设置低电压检测电路 (参见第二十三章 低电压检测电路)。**

**备注**　$V_{LVI}$: LVI 检测电压
　　　$V_{POC}$: POC 检测电压

<R>

**图 22-2. 由上电清零电路和低电压检测电路产生内部复位信号的时序 (2/2)**

**(2) 2.7 V/1.59 V POC 模式 (选项字节: POCMODE = 1)**




注　**1.** 保证操作的电压范围：1.8 V ≤ $V_{DD}$ ≤ 5.5 V。当供电电压下降时，为使状态低于 1.8 V 复位状态，可以使用低电压检测电路的复位功能，或向 RESET 引脚输入低电平。

**2.** 可以选择内部高速振荡时钟和一个高速系统时钟或一个子系统时钟作为 CPU 时钟。如果要使用 X1 时钟，可以通过 OSTC 确认是否经历了振荡稳定时间。而如果要使用 XT1 时钟，可以通过定时器功能来确认是否经历了振荡稳定时间。

**注意事项　1. 在释放复位状态后通过软件设置低电压检测电路 (参见第二十三章　低电压检测电路)。**

<R>

**2. 在电源电压抵达 1.59V (TYP.) 以后，需要一个 1.93 到 5.39 ms 的电压振荡稳定时间。如果在 1.93 ms 时间以内，电源电压由 1.59 V (TYP.) 上升到 2.7 V (TYP.)，则在复位处理之前，自动产生一个 0 到 5.39 ms 的电源振荡稳定时间。**

**备注**　$V_{LVI}$:　LVI 检测电压

$V_{POC}$:　POC 检测电压

## 22.4 上电清零电路使用注意事项

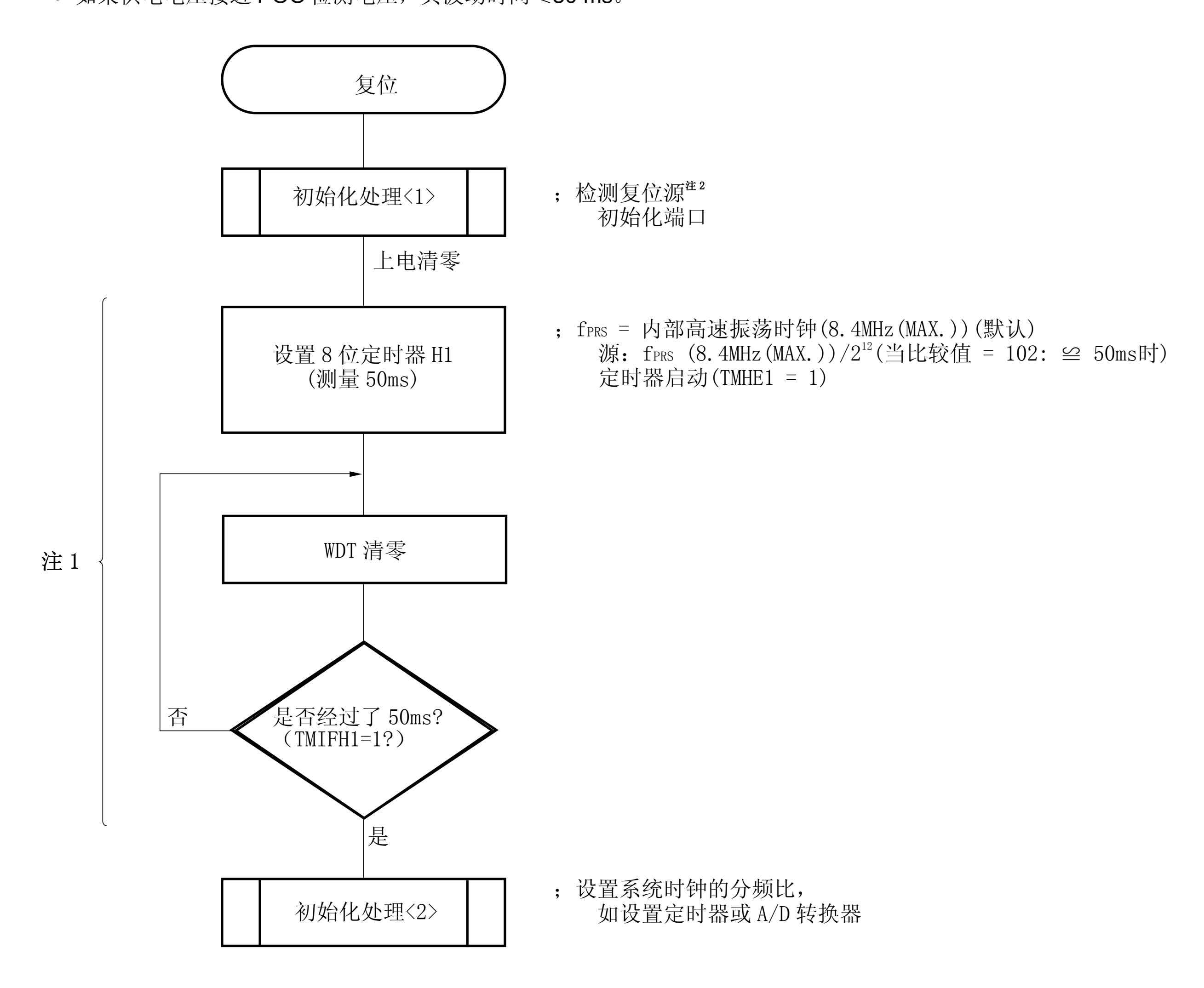
在系统中，如果供电电压 ($V_{DD}$) 接近 POC 检测电压 ($V_{POC}$) 时的一段时间内产生波动，则系统可能会反复复位和释放复位。在这种情况下，可采用下列方法任意设置从复位释放到微控制器的启动所需要的时间。

<方法>

在释放复位信号后，通过软件计数器（使用一个定时器）等待系统供电电压的波动期，然后对端口初始化。

**图 22-3. 复位释放后软件处理过程示例 (1/2)**

- 如果供电电压接近 POC 检测电压，其波动时间≤50 ms。

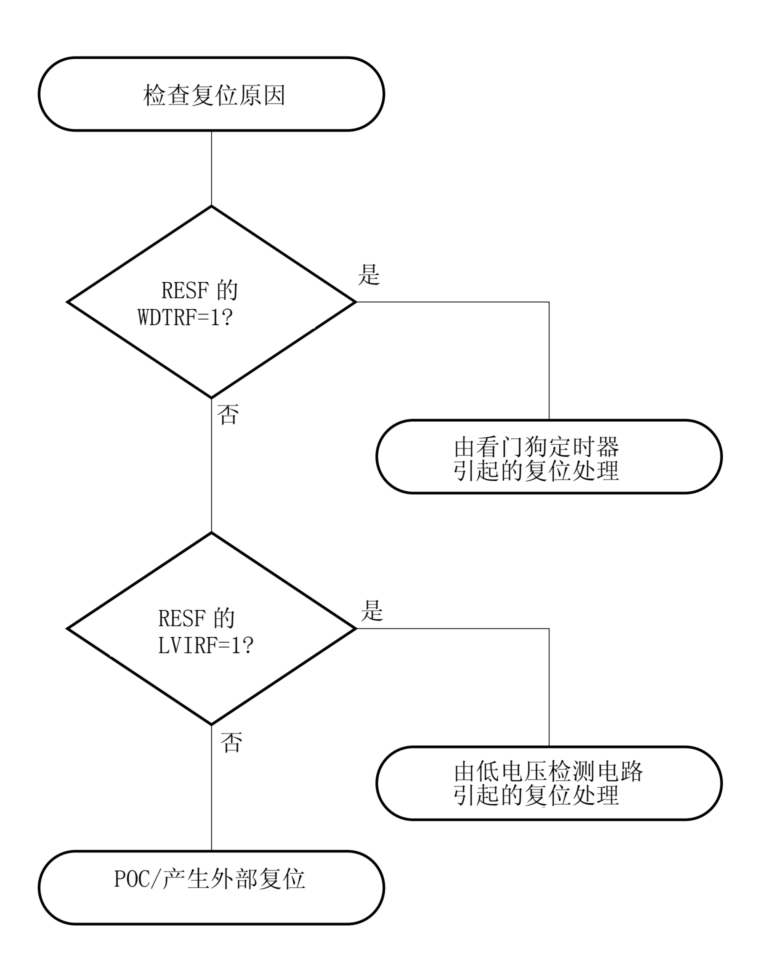



注　**1.** 如果在此期间再次产生复位，则不启动初始化操作<2>。

　　**2.** 流程图显示在下页。

图 22-3. 复位释放后软件处理过程示例 (2/2)

- 检查复位原因



# 第二十三章　低电压检测电路

## 23.1 低电压检测电路的功能

低电压检测电路 （LVI）有以下功能。

- 比较电源电压（$V_{DD}$）和检测电压（$V_{LVI}$），或者比较外部输入引脚的输入电压 (EXLVI) 与检测电压 ($V_{EXLVI}$ = 1.21 V (TYP.)：固定)，并产生内部中断信号或内部复位信号。
- 可由软件选择电源电压($V_{DD}$)或外部输入引脚的输入电压(EXLVI)。
- 可由软件选择复位或中断功能。
- 可由软件改变电源电压的检测等级 （16 个等级）。
- 在 STOP 模式下可操作。

<R> 以下，由软件来选择产生复位和中断信号。

| 选择供电电压($V_{DD}$)的等级检测<br>(LVISEL = 0) | | 选择从外部输入引脚(EXLVI)输入的电压的等级检测<br>(LVISEL = 1) | |
|---|---|---|---|
| 选择复位 (LVIMD = 1) | 选择中断(LVIMD = 0) | 选择复位(LVIMD = 1) | 选择中断(LVIMD = 0) |
| 当 $V_{DD} < V_{LVI}$ 时产生一个内部复位信号，当 $V_{DD} \geq V_{LVI}$ 时释放复位信号。 | 当 $V_{DD}$ 降至低于 $V_{LVI}$($V_{DD} < V_{LVI}$)时或者 $V_{DD}$ 升至高于 $V_{LVI}$ ($V_{DD} \geq V_{LVI}$) 时产生一个内部中断信号。 | 当 EXLVI < $V_{EXLVI}$ 时产生一个内部复位信号，并且在 EXLVI ≥ $V_{EXLVI}$ 时释放复位信号。 | 当 EXLVI 降至低于 $V_{EXLVI}$ (EXLVI < $V_{EXLVI}$)时或者 EXLVI 升至高于 $V_{EXLVI}$ (EXLVI ≥ $V_{EXLVI}$) 时产生一个内部中断信号。 |

**备注**　LVISEL: 低电压检测寄存器(LVIM)的第 2 位
　　　　LVIMD:　LVIM 的第 1 位

当低电压检测电路在操作时，通过读取低电压检测标志(LVIF: LVIM 的第 0 位)来检查供电电压或者外部输入引脚的输
<R> 入电压是否大于还是小于检测等级。

当低电压检测电路用作产生复位时，如果复位产生，复位控制标志寄存器(RESF)的第 0 位(LVIRF) 被置 1。如欲了解 RESF 的详细信息，请参见 **第二十一章　复位功能。**

## 23.2 低电压检测电路的配置

低电压检测电路的框图如图 23-1 所示。

**图 23-1. 低电压检测电路的框图**



## 23.3 控制低电压检测电路的寄存器

低电压检测电路由以下寄存器控制。

- 低电压检测寄存器 （LVIM）
- 低电压检测等级选择寄存器 （LVIS）
- 端口模式寄存器 12 (PM12)

**(1) 低电压检测寄存器 （LVIM）**

该寄存器用于设置低电压检测和操作模式。
可由 1 位或 8 位存储器操作指令设置该寄存器。
复位信号将 LVIM 清零（00H）。

### 图 23-2. 低电压检测寄存器（LVIM）的格式

地址：FFBEH　　　复位后：00H　R/W[注 1]

| 符号 | <7> | 6 | 5 | 4 | 3 | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| LVIM | LVION | 0 | 0 | 0 | 0 | LVISEL | LVIMD | LVIF |

| LVION[注 2, 3] | 允许低电压检测操作 |
|---|---|
| 0 | 禁止操作 |
| 1 | 允许操作 |

| LVISEL[注 2] | 电压检测的选择 |
|---|---|
| 0 | 检测供电电压($V_{DD}$)的等级 |
| 1 | 检测外部输入引脚的输入电压(EXLVI)的等级 |

<R>

| LVIMD[注 2] | 低电压检测操作模式(中断/复位)的选择 |
|---|---|
| 0 | • LVISEL = 0: 当供电电压（$V_{DD}$）降至低于检测电压（$V_{LVI}$）($V_{DD} < V_{LVI}$)时或者 $V_{DD}$ 大于 $V_{LVI}$ 时($V_{DD} \geq V_{LVI}$)产生内部中断信号。<br>• LVISEL = 1: 当外部输入引脚的输入电压(EXLVI) 降至低于检测电压($V_{EXLVI}$) (EXLVI < $V_{EXLVI}$)时或者 EXLVI 大于 $V_{EXLVI}$(EXLVI ≥ $V_{EXLVI}$)时产生中断信号。 |
| 1 | • LVISEL = 0: 当供电电压（$V_{DD}$） < 检测电压（$V_{LVI}$）时产生内部复位信号，并在 $V_{DD} \geq V_{LVI}$ 时释放复位信号。<br>• LVISEL = 1: 当外部输入引脚的输入电压(EXLVI) < 检测电压($V_{EXLVI}$)时，产生内部复位信号，并在 EXLVI ≥ $V_{EXLVI}$ 时释放复位信号。 |

| LVIF[注 4] | 低电压检测标志 |
|---|---|
| 0 | • LVISEL = 0: 供电电压（$V_{DD}$）≥ 检测电压（$V_{LVI}$）、或当禁止操作时<br>• LVISEL = 1: 外部输入引脚的输入电压(EXLVI) ≥ 检测电压($V_{EXLVI}$)，或当禁止操作时 |
| 1 | • LVISEL = 0: 供电电压（$V_{DD}$） < 检测电压（$V_{LVI}$）<br>• LVISEL = 1: 外部输入引脚的输入电压(EXLVI) < 检测电压($V_{EXLVI}$) |

注　**1.** 第 0 位只读。

**2.** 在产生除 LVI 复位以外的复位时，LVION、LVIMD 和 LVISEL 被清零(0)。在产生 LVI 复位时这些位不被清零。

**3.** 当 LVION=1 时，LVI 电路中的比较器开始操作。从设置 LVION=1 到 LVIF 确定电压，使用软件等待操作稳定时间(10 $\mu$s (MAX.))。

**4.** 当 LVION = 1 且 LVIMD = 0 时，输出 LVIF 的值作为中断请求 INTLVI 的中断源。

注意事项　**1. 在执行以下任一过程后，停止 LVI。**

- **当使用 8 位操作指令时： 00H → LVIM。**
- **当使用 1 位存储器操作指令时：对 LVION 清零。**

**2. 外部输入引脚的输入电压(EXLVI)必须为： EXLVI < $V_{DD}$。**

**(2)　低电压检测等级选择寄存器 (LVIS)**

该寄存器用于选择低电压检测等级。

可由 1 位或 8 位存储器操作指令设置该寄存器。

复位信号将 LVIS 清零（00H）。

**图 23-3.** 低电压检测等级选择寄存器 （**LVIS**）的格式

地址：FFBFH　　复位后：00H　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| LVIS | 0 | 0 | 0 | 0 | LVIS3 | LVIS2 | LVIS1 | LVIS0 |

| LVIS3 | LVIS2 | LVIS1 | LVIS0 | 检测等级 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $V_{LVI0}$ (4.24 V ±0.1 V) |
| 0 | 0 | 0 | 1 | $V_{LVI1}$ (4.09 V ±0.1 V) |
| 0 | 0 | 1 | 0 | $V_{LVI2}$ (3.93 V ±0.1 V) |
| 0 | 0 | 1 | 1 | $V_{LVI3}$ (3.78 V ±0.1 V) |
| 0 | 1 | 0 | 0 | $V_{LVI4}$ (3.62 V ±0.1 V) |
| 0 | 1 | 0 | 1 | $V_{LVI5}$ (3.47 V ±0.1 V) |
| 0 | 1 | 1 | 0 | $V_{LVI6}$ (3.32 V ±0.1 V) |
| 0 | 1 | 1 | 1 | $V_{LVI7}$ (3.16 V ±0.1 V) |
| 1 | 0 | 0 | 0 | $V_{LVI8}$ (3.01 V ±0.1 V) |
| 1 | 0 | 0 | 1 | $V_{LVI9}$ (2.85 V ±0.1 V) |
| 1 | 0 | 1 | 0 | $V_{LVI10}$ (2.70 V ±0.1 V) |
| 1 | 0 | 1 | 1 | $V_{LVI11}$ (2.55 V ±0.1 V) |
| 1 | 1 | 0 | 0 | $V_{LVI12}$ (2.39 V ±0.1 V) |
| 1 | 1 | 0 | 1 | $V_{LVI13}$ (2.24 V ±0.1 V) |
| 1 | 1 | 1 | 0 | $V_{LVI14}$ (2.08 V ±0.1 V) |
| 1 | 1 | 1 | 1 | $V_{LVI15}$ (1.93 V ±0.1 V) |

注意事项　**1.** 第 **4** ～ **7** 位必须清零。

**2.** **LVI** 操作期间不要修改 **LVIS** 的值。

**3.** 当检测到外部输入引脚的输入电压**(EXLVI)**时，检测电压**($V_{EXLVI}$ = 1.21 V (TYP.))**恒定。因此，不必设置 **LVIS**。

**(3)　端口模式寄存器 12 (PM12)**

当使用 P120/EXLVI/INTP0 引脚进行外部低电压检测电压输入时，设置 PM120=1。此时，P120 的输出锁存器的值可以为 0 或 1。

可由 1 位或 8 位存储器操作指令设置 PM12。

复位信号将 PM12 设置为 FFH。

**图 23-4. 端口模式寄存器 12 (PM12)的格式**

地址：FF2CH　　复位后：FFH　R/W

| 符号： | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM12 | 1 | 1 | 1 | PM124 | PM123 | PM122 | PM121 | PM120 |

| PM12n | P12n 引脚 I/O 模式的选择 (n = 0 ～ 4) |
|---|---|
| 0 | 输出模式 (输出缓冲器打开) |
| 1 | 输入模式 (输出缓冲器关闭) |

## 23.4 低电压检测电路的操作

低电压检测电路能够用于以下两种模式。

<R> **(1)　用于复位 (LVIMD = 1)**

- 如果 LVISEL = 0，比较供电电压($V_{DD}$)与检测电压($V_{LVI}$)，当 $V_{DD} < V_{LVI}$ 时产生内部复位信号，而当 $V_{DD} \geq V_{LVI}$ 时释放内部复位。
- 如果LVISEL = 1，比较外部输入引脚的输入电压(EXLVI)和检测电压($V_{EXLVI}$ = 1.21 V (TYP.))，在EXLVI < $V_{EXLVI}$ 时产生内部复位信号，而在EXLVI ≥ $V_{EXLVI}$时释放内部复位。

<R> **(2)　用于中断 (LVIMD = 0)**

- 如果 LVISEL = 0，比较供电电压($V_{DD}$)和检测电压($V_{LVI}$)，在 $V_{DD}$ 降至低于 $V_{LVI}$（$V_{DD} < V_{LVI}$）时或者 $V_{DD}$ 高于 $V_{LVI}$($V_{DD} \geq V_{LVI}$)时产生中断信号(INTLVI)。
- 如果LVISEL = 1，比较比较外部输入引脚的输入电压(EXLVI)和检测电压($V_{EXLVI}$ = 1.21 V (TYP.))，在EXLVI < $V_{EXLVI}$时产生中断信号(INTLVI)。当EXLVI降至低于$V_{EXLVI}$ (EXLVI < $V_{EXLVI}$) 时或者EXLVI 高于$V_{EXLVI}$ 时(EXLVI ≥ $V_{EXLVI}$)，产生一个中断信号 (INTLVI)。

当低电压检测电路在操作时，通过读取低电压检测标志(LVIF: LVIM 的第 0 位)来检查供电电压或者外部输入引脚的输入电压是大于还是小于检测等级。<R>

**备注**　LVIMD:  低电压检测寄存器(LVIM)的第 1 位
　　　LVISEL: LVIM 的第 2 位

### 23.4.1 当用于复位

**(1) 当检测供电电压($V_{DD}$)等级时**

- 启动操作时

  <1>　屏蔽 LVI 中断 （LVIMK = 1）。
  <2>　将低电压检测寄存器(LVIM)的第 2 位(LVISEL)清零(检测供电电压($V_{DD}$)的等级)(默认值)。
  <3>　使用低电压检测等级选择寄存器（LVIS）的第 3～0 位（LVIS3 ～ LVIS0）设置检测电压。
  <4>　设置 LVIM 的第 7 位 （LVION）=1 （允许 LVI 的操作）。
  <5>　使用软件等待操作稳定时间(10 μs (MAX.))。
  <6>　等待直至根据 LVIM 的第 0 位(LVIF)检测到电压(供电电压($V_{DD}$) ≥ 检测电压($V_{LVI}$))。
  <7>　设置 LVIM 的第 1 位 （LVIMD）=1 （当供电电压（$V_{DD}$） < 检测电压（$V_{LVI}$）时产生内部复位信号）。

  图 23-5 显示了由低电压检测电路产生的内部复位信号的时序。该时序图中的标号与上面的<1> ～ <7>对应。

  **注意事项　1.　必须执行过程<1>。当 LVIMK = 0 时，在过程<4>之后可能立即会产生中断。**
  **2.　当 LVIMD=1 时如果供电电压（$V_{DD}$） ≥ 检测电压（$V_{LVI}$），则不会产生内部复位信号。**

- 当停止操作时

  必须执行以下过程之一。

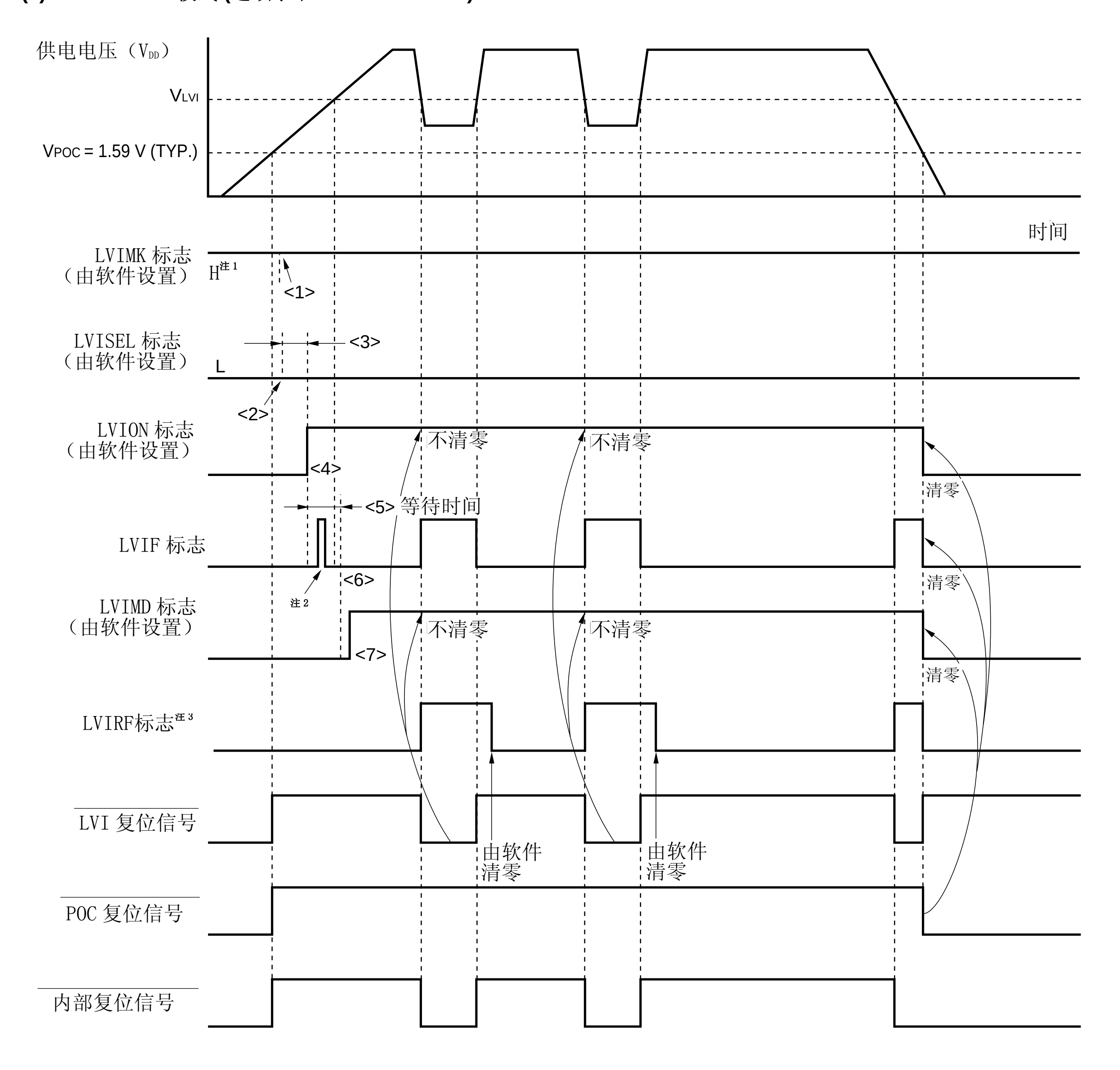
  - 当使用 8 位存储器操作指令时:
    写 00H → LVIM

  - 当使用 1 位存储器操作指令时:
    依次将 LVIMD 和 LVION 清零。

图 **23-5.** 低电压检测电路内部复位信号产生时序

**(检测供电电压(VDD)等级) (1/2)**

**(1) 1.59 V POC 模式 (选项字节: POCMODE = 0)**




注 **1.** 由复位信号产生将 LVIMK 标志设置为“1”。

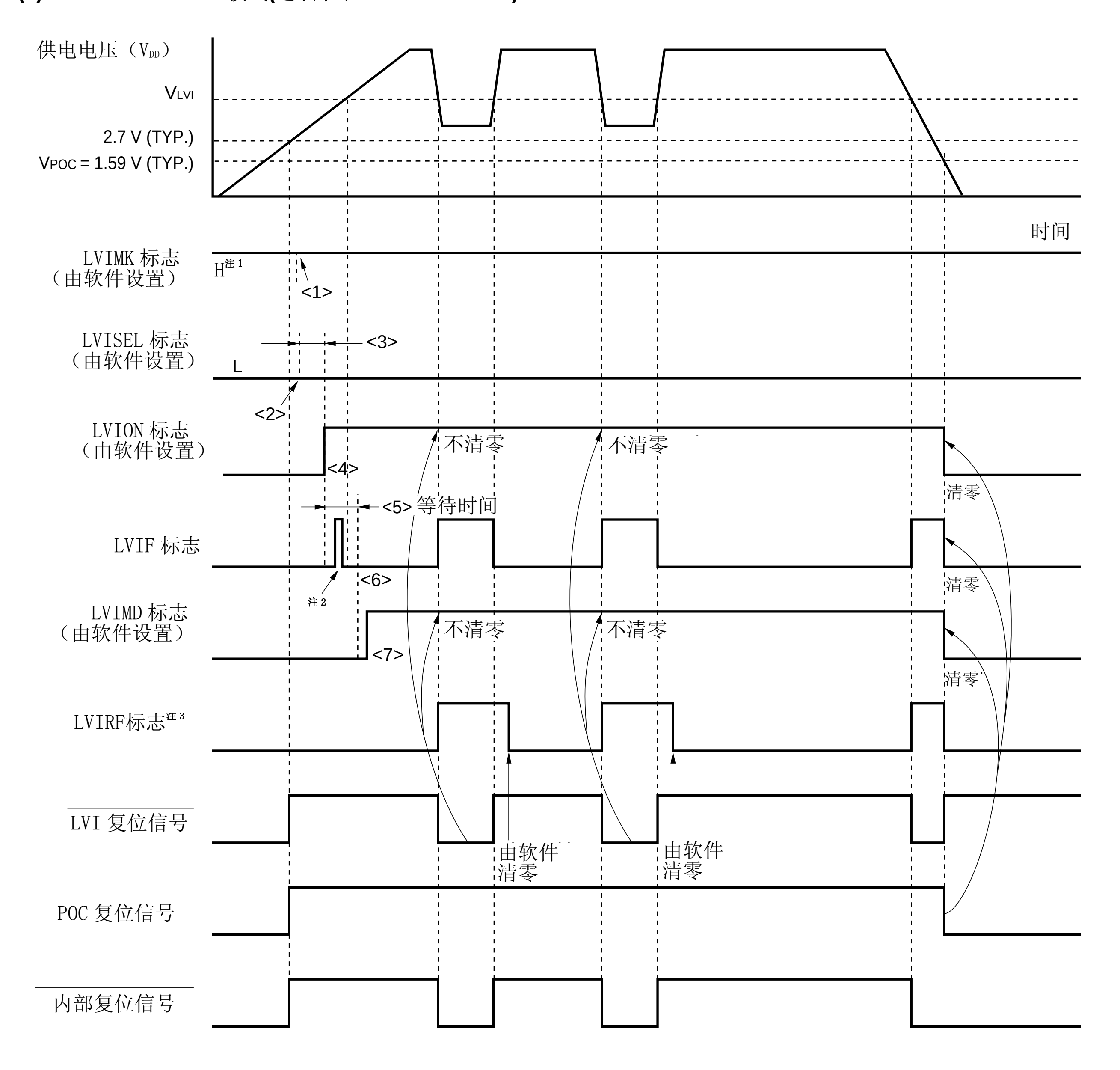
**2.** 可以设置 LVIF 标志为(1)。

**3.** LVIRF 是复位控制标志寄存器（RESF）的第 0 位。需要了解 RESF 的详细内容，可参见**第二十一章 复位功能**。

备注 图 23-5 中的<1> ～ <7>与 **23.4.1(1) 当检测供电电压(VDD)等级时**中“当启动操作时”的描述<1> ～ <7>相对应。

**图 23-5. 低电压检测电路内部复位信号产生时序**
**(检测供电电压($V_{DD}$)等级) (2/2)**

**(2) 2.7 V/1.59 V POC 模式(选项字节: POCMODE = 1)**




注　**1.** 由复位信号产生将 LVIMK 标志设置为“1”。

**2.** 可以设置 LVIF 标志为(1)。

**3.** LVIRF 是复位控制标志寄存器（RESF）的第 0 位。需要了解 RESF 的详细内容，可参见**第二十一章　复位功能**。

**备注**　图 23-5 中的<1> ～ <7>与 **23.4.1(1) 当检测供电电压($V_{DD}$)等级时**中“当启动操作时”的描述<1> ～ <7>相对应。

**(2) 当检测外部输入引脚的输入电压(EXLVI)等级时**

- 当启动操作时

<1>　屏蔽 LVI 中断 (LVIMK = 1)。

<2>　设置低电压检测寄存器(LVIM)的第 2 位(LVISEL)=1 (检测外部输入引脚的输入电压 (EXLVI) 等级)。

<3>　设置 LVIM 的第 7 位 （LVION）=1 （允许 LVI 的操作）。

<4>　使用软件等待操作稳定时间(10 μs (MAX.)[注])。

<5>　等待直至根据 LVIM 的第 0 位(LVIF)检测到：外部输入引脚的输入电压(EXLVI) ≥ 检测电压($V_{EXLVI}$ = 1.21 V (TYP.))

<6>　设置 LVIM 的第 1 位(LVIMD)=1(当外部输入引脚的输入电压(EXLVI) < 检测电压($V_{EXLVI}$ = 1.21 V (TYP.))时产生内部复位信号)。

图 23-6 显示了由低电压检测电路产生的内部复位信号的时序。该时序图中的标号与上面的<1> ～ <6>对应。

注意事项 **1. 必须执行<1>。当 LVIMK = 0 时，在完成<3>后可能立即会产生中断。**

**2. 如果在 LVIMD=1 时，外部输入引脚的输入电压(EXLVI) ≥ 检测电压($V_{EXLVI}$ = 1.21 V (TYP.))，则不会产生内部复位信号。**

**3. 必须保证外部输入引脚的输入电压(EXLVI) ：EXLVI < $V_{DD}$**

- 当停止操作时

必须执行以下过程之一。

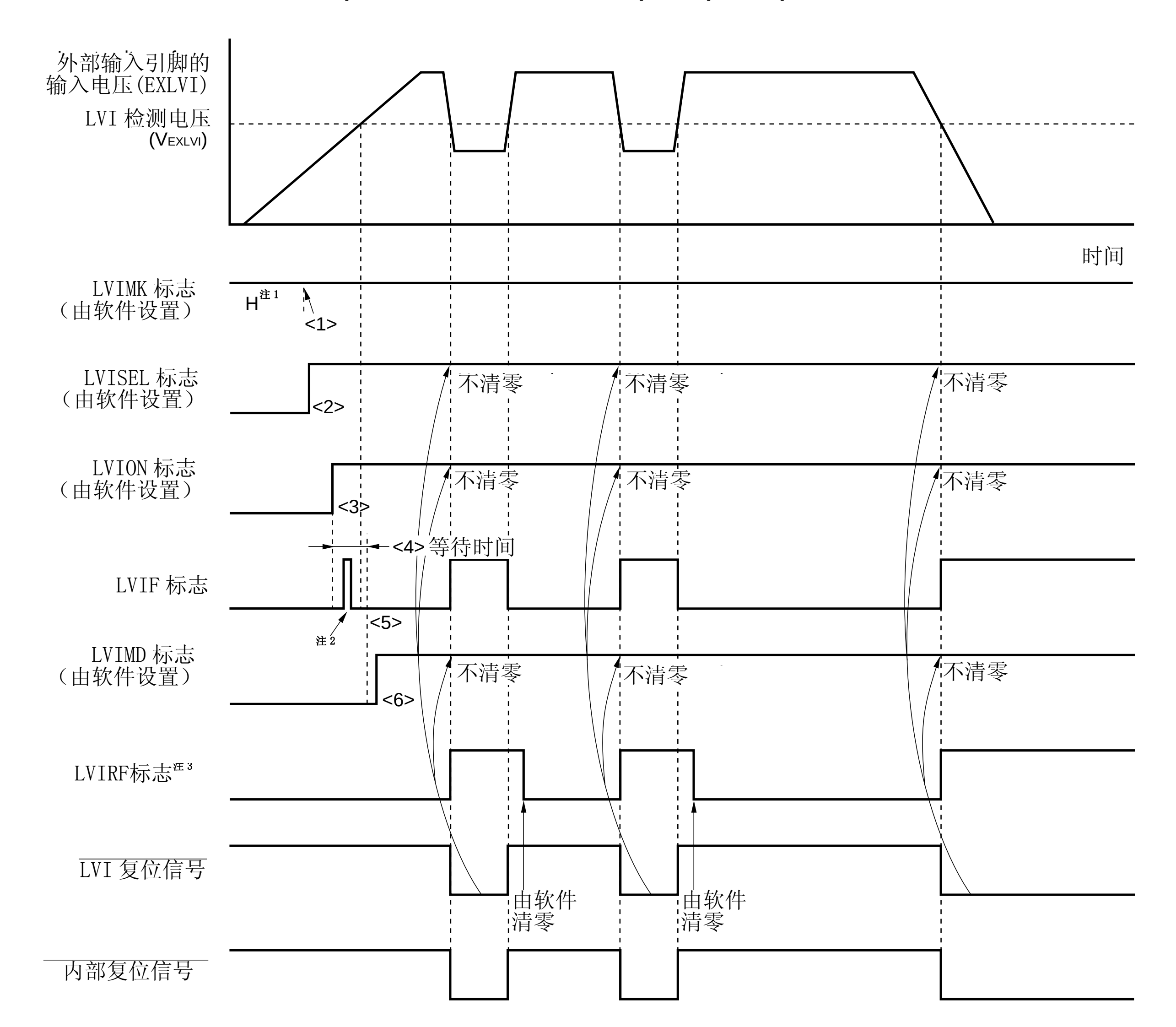
- 当使用 8 位存储器操作指令:

  写 00H →LVIM

- 当使用 1 位存储器操作指令:

  先对 LVIMD 清零(0)，再对 LVION 清零(0)。

**图 23-6. 低电压检测电路内部复位信号产生时序**
**(检测外部输入引脚的输入电压(EXLVI)的等级)**




注　**1.** 由复位信号的产生设置 LVIMK 标志为 “1”。

**2.** 可以设置 LVIF 标志为(1)。

**3.** LVIRF 为复位控制标志寄存器(RESF)的第 0 位。如需了解 RESF 的详细信息，可参见**第二十一章　复位功能**。

备注　图 23-6 中的<1> ～ <6>与 **23.4.1 (2) 当检测外部输入引脚的输入电压(EXLVI)等级时**中“当启动操作时”的描述<1> ～ <6>相对应。

### 23.4.2 当用于中断时

**(1) 当检测供电电压($V_{DD}$)等级时**

- 当启动操作时

  <1>　屏蔽 LVI 中断 （LVIMK = 1）
  <2>　将低电压检测寄存器(LVIM)的第 2 位(LVISEL)清零(0)(检测供电电压($V_{DD}$)等级)(默认值)。
  <3>　使用低电压检测等级选择寄存器（LVIS）的第 3～0 位（LVIS3 ～ LVIS0）设置检测电压。
  <4>　设置 LVIM 的第 7 位 （LVION）=1 （允许 LVI 的操作）。
  <5>　使用软件等待操作稳定时间(10 μs (MAX.))。
<R>　<6>　在 LVIM 的第 0 位(LVIF)，当检测 $V_{DD}$ 的下降沿时，确定“供电电压($V_{DD}$) ≥ 检测电压($V_{LVI}$)”，或者当检测 $V_{DD}$ 的上升沿时，确定“供电电压($V_{DD}$) < 检测电压($V_{LVI}$)” 。
  <7>　将 LVI 的中断请求标志（LVIIF）清零(0)。
  <8>　释放 LVI 的中断屏蔽标志 （LVIMK）。
  <9>　将 LVIM 的第 1 位(LVIMD)清零(0)(当供电电压($V_{DD}$) < 检测电压($V_{LVI}$)时产生中断信号)(默认值)。
  <10> 执行 EI 指令 （当使用向量中断时）。

  图 23-7 显示了由低电压检测电路产生的中断信号的时序。该时序图中的标号与上面的<1> ～ <9>对应。

- 当停止操作时

  必须执行以下过程之一。

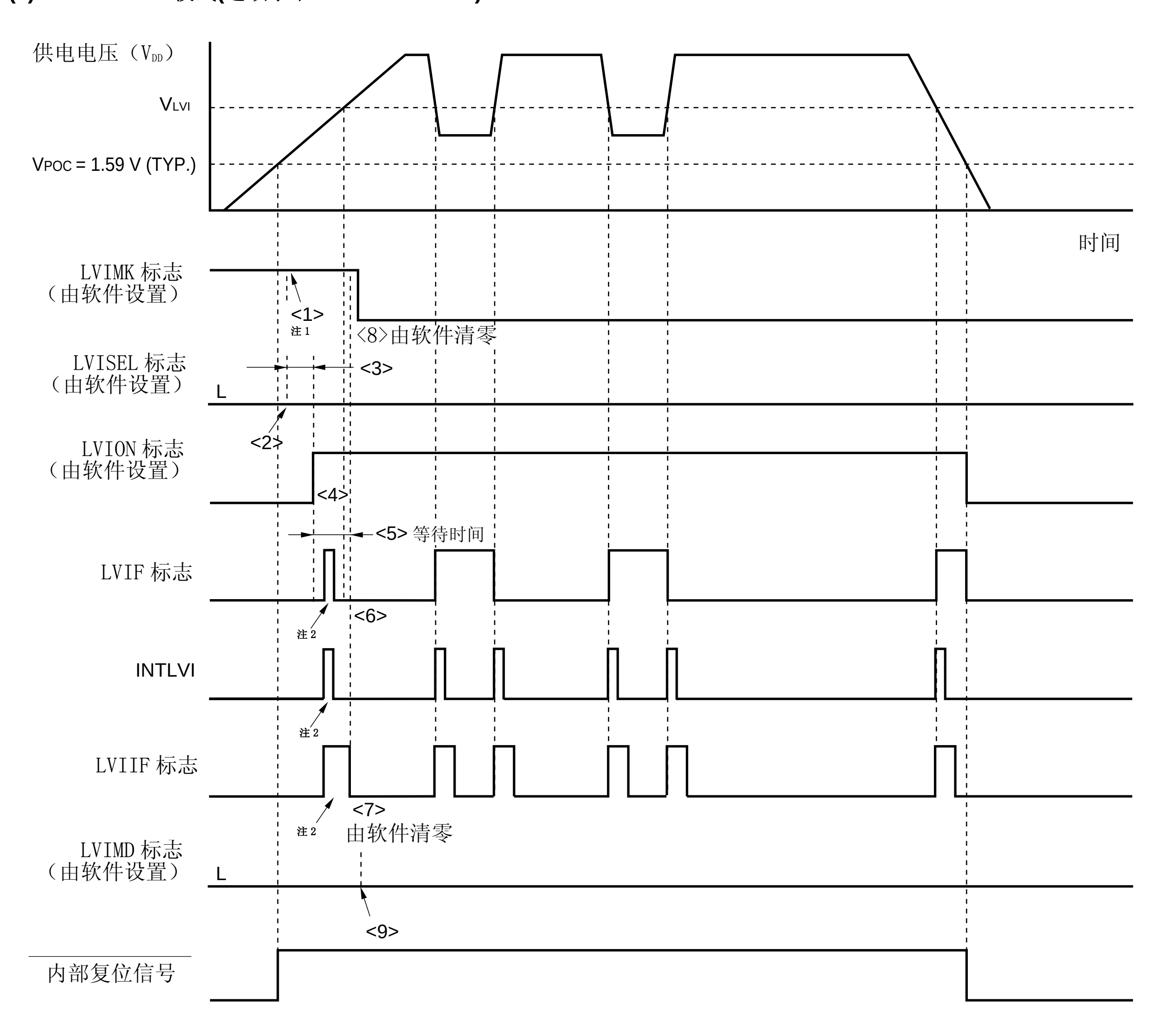
  - 当使用 8 位存储器操作指令:
    写 00H →LVIM

  - 当使用 1 位存储器操作指令:
    对 LVION 清零（0）

<R>

图 **23-7.** 低电压检测电路中断信号产生时序
**(检测供电电压($V_{DD}$)的等级) (1/2)**

**(1) 1.59 V POC 模式(选项字节: POCMODE = 0)**




注　**1.**　可由复位信号的产生设置 LVIMK 标志为“1”。
　　**2.**　产生中断请求信号(INTLVI)，且可以设置 LVIF 和 LVIIF 标志为(1)。

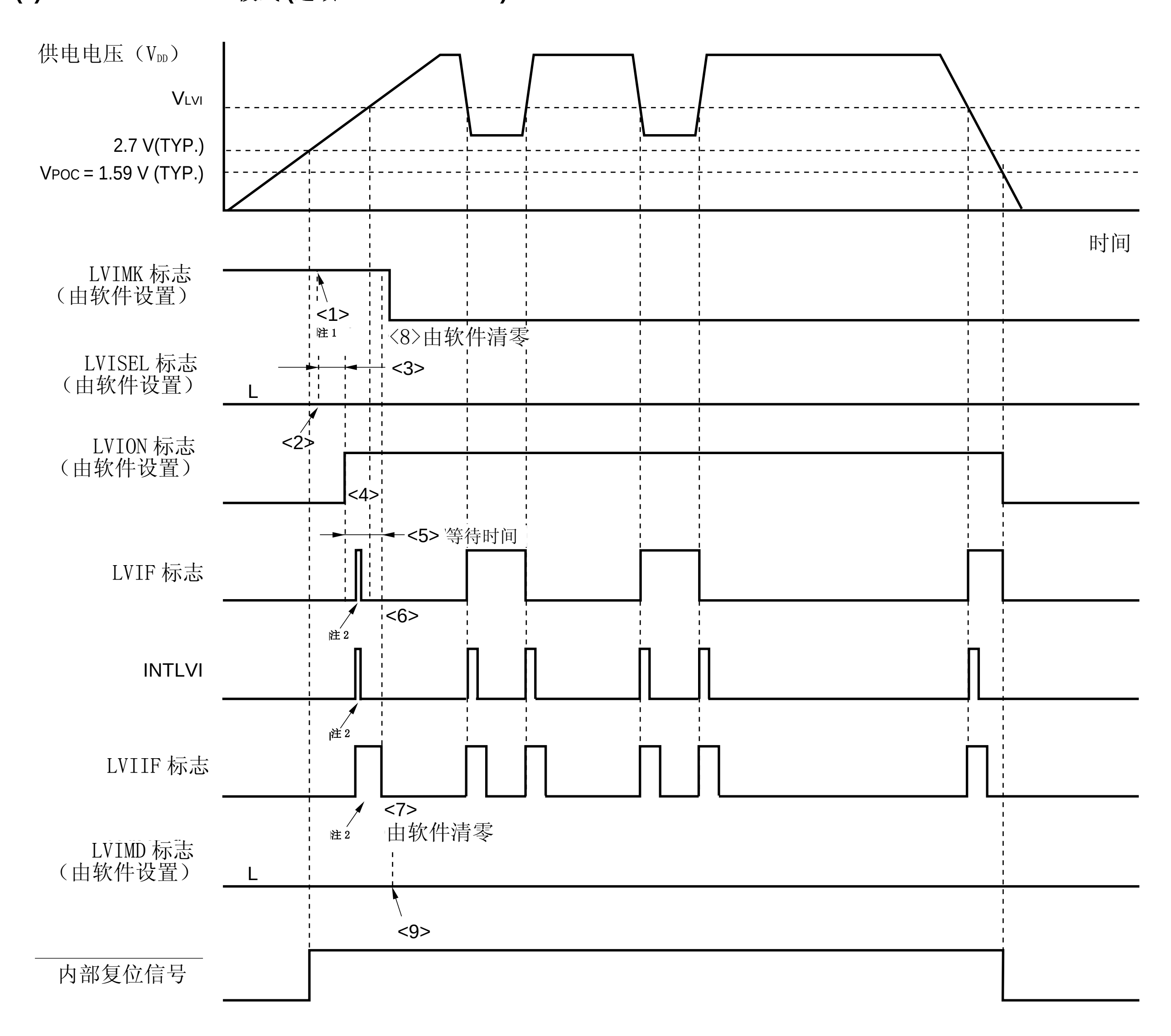
备注　图 23-7 中的<1> ～ <9>与 **23.4.2 (1) 当检测供电电压($V_{DD}$)等级时**中“当启动操作时”的描述<1> ～ <9>相对应。

<R>

图 **23-7.** 低电压检测电路中断信号产生时序

**(检测供电电压($V_{DD}$)的等级) (2/2)**

**(2) 2.7 V/1.59 V POC 模式 (选项: POCMODE = 1)**




注　**1.** 可由复位信号的产生设置 LVIMK 标志为"1"。

　　**2.** 产生中断请求信号(INTLVI)，且可以设置 LVIF 和 LVIIF 标志为(1)。

备注　图 23-7 中的<1> ～ <9>与 **23.4.2 (1) 当检测供电电压($V_{DD}$)等级时**中"当启动操作时"的描述<1> ～ <9>相对应。

**(2)　当检测外部输入引脚的输入电压(EXLVI)等级时**

- 当启动操作时

<1>　屏蔽 LVI 中断 （LVIMK = 1）

<2>　设置低电压检测寄存器(LVIM)的第 2 位(LVISEL)=1(检测外部输入引脚的输入电压(EXLVI)等级)。

<3>　设置 LVIM 的第 7 位 （LVION）=1 （允许 LVI 的操作）。

<4>　使用软件等待操作稳定时间(10 μs (MAX.))。

<R>　<5>　在 LVIM 的第 0 位(LVIF)，当检测 EXLVI 的下降沿时，确定“外部输入引脚的输入电压(EXLVI) ≥ 检测电压($V_{EXLVI}$ = 1.21 V (TYP.))”，或者当检测 EXLVI 的上升沿时，确定“外部输入引脚的输入电压(EXLVI)< 检测电压($V_{EXLVI}$ = 1.21 V (TYP.))”。

<6>　将 LVI 的中断请求标志（LVIIF）清零(0)。

<7>　释放 LVI 的中断屏蔽标志 （LVIMK）。

<8>　将 LVIM 的第 1 位(LVIMD)清零(0)(当供电电压($V_{DD}$) < 检测电压($V_{LVI}$)时产生中断信号)(默认值)。

<9>　执行 EI 指令 （当使用向量中断时）。

图 23-8 显示了由低电压检测电路产生的中断信号的时序。该时序图中的标号与上面的<1> ～ <8>对应。

**注意事项　外部输入引脚的输入电压(EXLVI)必须满足以下条件：EXLVI < $V_{DD}$**

- 当停止操作时

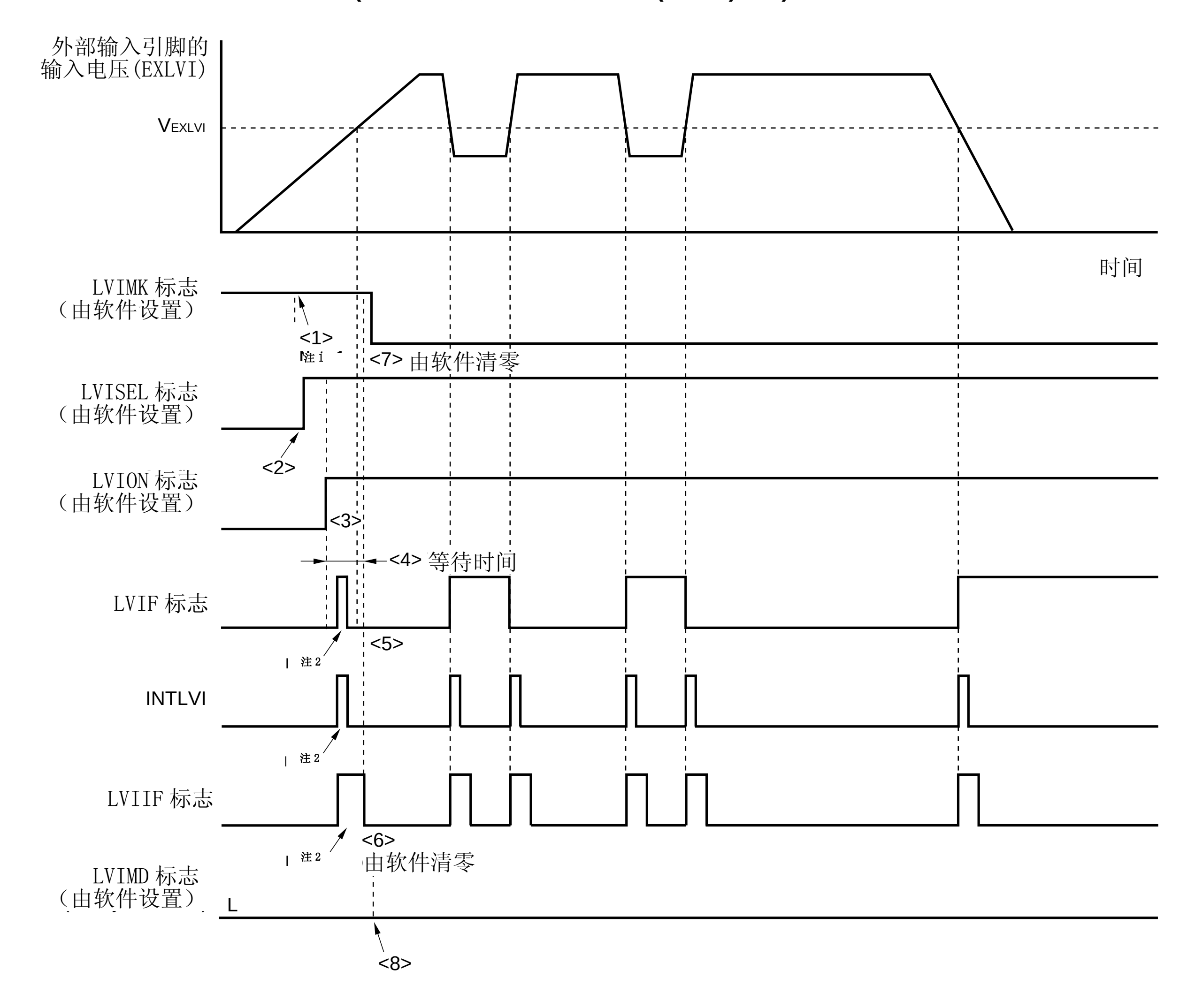
必须执行以下过程之一。

  - 当使用 8 位存储器操作指令:
    写 00H →LVIM

  - 当使用 1 位存储器操作指令:
    对 LVION 清零（0）

<R>

**图 23-8. 低电压检测电路中断信号产生时序**
**(检测外部输入引脚的输入电压(EXLVI)等级)**




注　**1.**　可由复位信号产生设置 LVIMK 标志为“1”。
　　**2.**　产生中断请求信号(INTLVI)，且可以设置 LVIF 和 LVIIF 标志为(1)。

备注　图 23-8 中的<1> ～ <8>与 **23.4.2 (2) 当检测外部输入引脚(EXLVI)的输入电压等级时**中“当启动操作时”的描述<1> ～ <8>相对应。

## 23.5 低电压检测电路的注意事项

在系统中，如果供电电压（$V_{DD}$）在接近 LVI 检测电压（$V_{LVI}$）的一段时间内产生波动，这时根据低电压检测电路的使用情况进行如下操作。

**(1)** 用于复位

系统可能会反复进行复位和释放复位。

在这种情况下，可采用下面的方法（1）任意设置从复位释放到微控制器的启动所经历的时间。

**(2)** 用于中断

可能会频繁产生中断请求。可采用方法（2）中的(b)。

<方法>

**(1)** 用于复位

在释放复位信号后，通过软件计数器（使用一个定时器）等待供电电压的波动期，然后对端口初始化(参见图 **23-9**)。

**(2)** 用于中断

<R> (a) 在 LVI 中断的服务程序中通过使用低电压检测寄存器（LVIM）的第 0 位(LVIF)，当检测 $V_{DD}$ 的下降沿时，确定“供电电压($V_{DD}$) ≥ 检测电压($V_{LVI}$)”，或者当检测 $V_{DD}$ 的上升沿时，确定“供电电压($V_{DD}$)< 检测电压($V_{LVI}$)”，把中断请求标志寄存器 0L(IF0L)的第 0 位(LVIIF)清零。

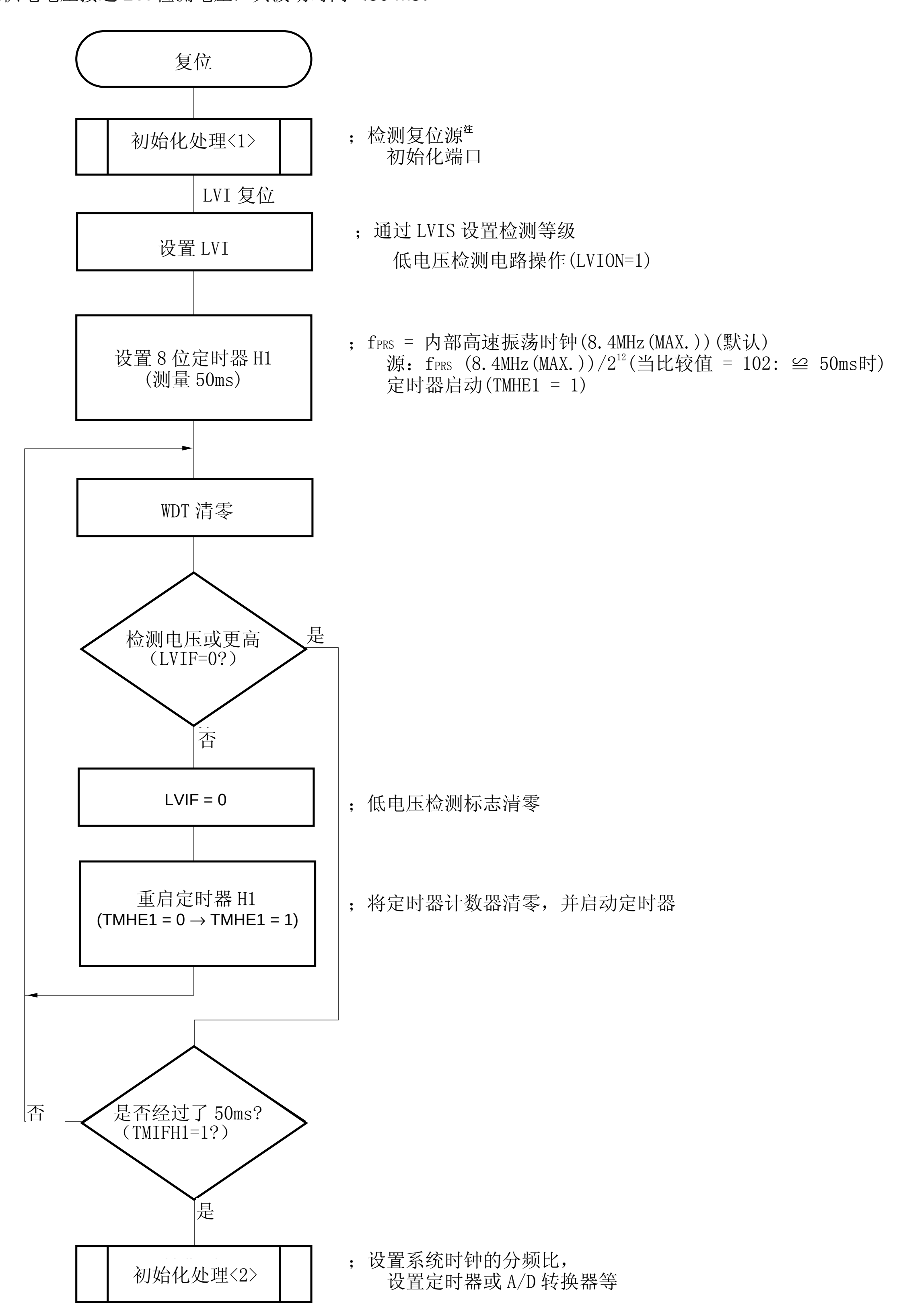
<R> (b) 在系统中，如果供电电压（$V_{DD}$）在接近 LVI 检测电压（$V_{LVI}$）的一段时间内产生波动，等待供电电压波动期间，使用 LVIF 标志，并且把 LVIIF 标志清零，当检测 $V_{DD}$ 的下降沿时，确定“供电电压($V_{DD}$) ≥ 检测电压($V_{LVI}$)”，或者当检测 $V_{DD}$ 的上升沿时，确定“供电电压($V_{DD}$)< 检测电压($V_{LVI}$)”。

**备注**　如果设置低电压检测寄存器(LVIM)的第 2 位(LVISEL)为“1”，则对上述方法作如下修改。

- 供电电压($V_{DD}$) → 外部输入引脚的输入电压(EXLVI)
- 检测电压($V_{LVI}$) → 检测电压($V_{EXLVI}$ = 1.21 V[注])

**图 23-9.  复位释放后软件处理过程示例 (1/2)**

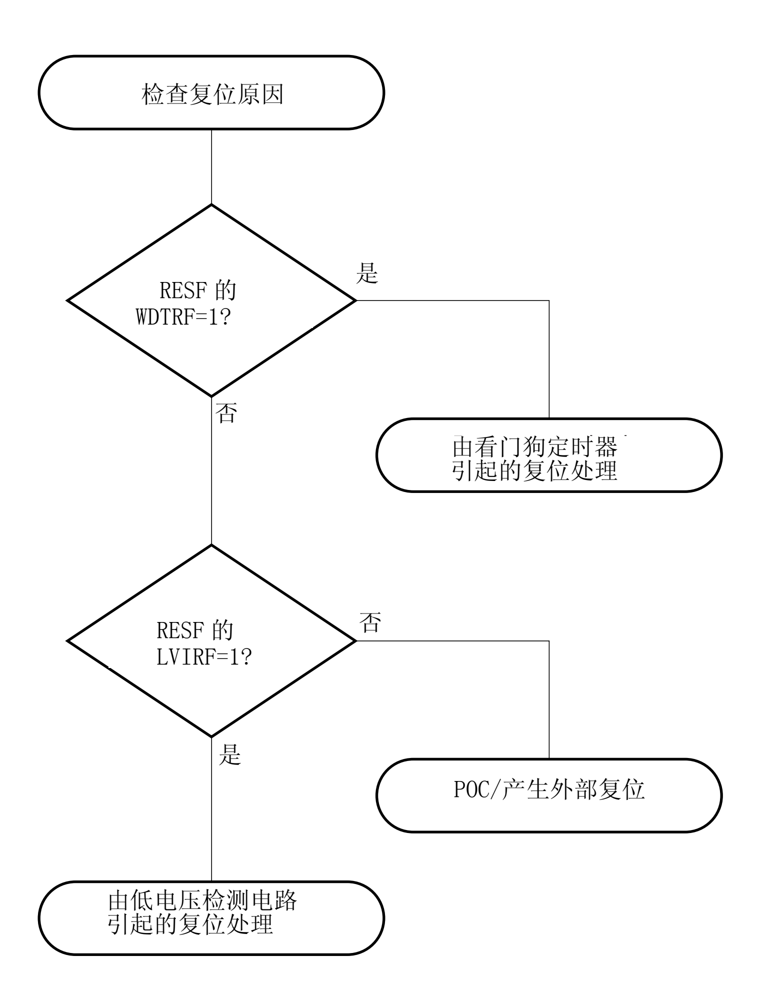
- 如果供电电压接近 LVI 检测电压，其波动时间≤50 ms。




**注**　　流程图显示在下页。

**图 23-9. 复位释放后软件处理过程示例 (2/2)**

- 检查复位原因



## 24.1 选项字节的功能

78K0/KC2 中 Flash 存储器的 0080H ～ 0084H 区域为选项字节。当打开电源或从复位状态重启设备时，设备自动参考选项字节，并设置指定的功能。使用该产品时，必须使用选项字节设置以下几项功能。

在自编程期间使用引导交换功能时，将 0080H ～ 0084H 切换到 1080H ～ 1084H。因此，事先将 0080H ～ 0084H 和 1080H ～ 1084H 设置为相同的值。

**注意事项　确保将地址 0082H 和 0083H 的内容清零 (0082H/1082H 和 0083H/1083H，当使用启动交换功能时)。**

**(1) 0080H/1080H**

○ 内部低速振荡器的操作
- 可由软件停止
- 不能停止

○ 看门狗定时器间隔时间的设置

○ 看门狗定时器计数器的操作
- 允许计数器的操作
- 禁止计数器的操作

○ 看门狗定时器窗口打开周期的设置

<R> **注意事项　因为地址 0080H 和 1080H 在启动交换操作中切换，所以要将 0080H 的值赋值给 1080H。**

**(2) 0081H/1081H**

○ 选择 POC 模式
- 2.7 V/1.59 V POC 模式操作期间 (POCMODE = 1)
  上电时该设备处于复位状态，直至供电电压达到 2.7 V (TYP.)。当电压超过 2.7 V (TYP.)时设备从复位状态释放。此后在 2.7 V 不检测 POC，但在 1.59 V (TYP.)检测 POC。
  <R> 如果上电后当速率低于 0.5 V/ms (MIN.)时供电电压上升到 1.8 V，建议使用 2.7 V/1.59 V POC 模式。
- 1.59 V POC 模式操作期间 (POCMODE = 0)
  上电时该设备处于复位状态，直至供电电压达到 1.59 V (TYP.)。当电压超过 1.59 V (TYP.)时设备从复位状态释放。此后采用与上电时相同的方法在 1.59 V (TYP.)检测 POC。

<R> **注意事项　POCMODE 只能通过专用 Flash 编程器写入。在自编程中(此时，设定为 1.59V POC 模式(缺省))，自编程或者启动交换操作都不能用于设定 POCMODE 的值。以上，因为在启动交换操作中 1081H 的值将拷贝到 0081H 中，所以推荐使用启动交换功能时，先将 0081H 的值赋给 1081H。**

**(3)　0084H/1084H**

○ 片上调试操作控制

- 禁止片上调试操作
- 一旦证明片上调试安全 ID 失效时，允许片上调试操作和擦除 Flash 存储器数据。
- 一旦证明片上调试安全 ID 失效时，允许片上调试操作，但不擦除 Flash 存储器数据。

注意事项　**1.** 对于没有配备片上调试功能的产品**(μPD78F0511，78F0512，78F0513，78F0514，**和 **78F0515)**，必须将 **00H** 赋给 **0084H** 单元**(**禁止片上调试操作**)**。由于在引导交换时 **0084H** 和 **1084H** 要相互切换，因此也要对 **1084H** 设置 **00H**。

**2.** 对于具有片上调试功能的产品**(μPD78F0513D，78F0515D)**，如果要使用它的片上调试功能，需要对 **0084H** 单元赋值 **02H** 或 **03H**。由于在引导交换时 **0084H** 和 **1084H** 要相互切换，因此也要对 **1084H** 设置与 **0084H** 相同的值。

## 24.2 选项字节的格式

选项字节格式如下。

图 **24-1.** 选项字节的格式 **(1/2)**

地址: 0080H/1080H[注]

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | WINDOW1 | WINDOW0 | WDTON | WDCS2 | WDCS1 | WDCS0 | LSROSC |

| WINDOW1 | WINDOW0 | 看门狗定时器窗口打开周期 |
|---|---|---|
| 0 | 0 | 25% |
| 0 | 1 | 50% |
| 1 | 0 | 75% |
| 1 | 1 | 100% |

| WDTON | 看门狗定时器计数器的操作控制/非法访问检测 |
|---|---|
| 0 | 禁止计数器操作 (复位后停止计数)，禁止非法访问检测操作 |
| 1 | 允许计数器操作 (复位后开始计数)，允许非法访问检测操作 |

| WDCS2 | WDCS1 | WDCS0 | 看门狗定时器溢出时间 |
|---|---|---|---|
| 0 | 0 | 0 | $2^{10}/f_{RL}$ (3.88 ms) |
| 0 | 0 | 1 | $2^{11}/f_{RL}$ (7.76 ms) |
| 0 | 1 | 0 | $2^{12}/f_{RL}$ (15.52 ms) |
| 0 | 1 | 1 | $2^{13}/f_{RL}$ (31.03 ms) |
| 1 | 0 | 0 | $2^{14}/f_{RL}$ (62.06 ms) |
| 1 | 0 | 1 | $2^{15}/f_{RL}$ (124.12 ms) |
| 1 | 1 | 0 | $2^{16}/f_{RL}$ (248.24 ms) |
| 1 | 1 | 1 | $2^{17}/f_{RL}$ (496.48 ms) |

| LSROSC | 内部低速振荡器操作 |
|---|---|
| 0 | 可由软件停止 (当将 1 写入 RCM 寄存器的第 0 位(LSRSTOP)时停止操作) |
| 1 | 不能停止 (即使将 1 写入 LSRSTOP 位也不停止操作) |

注　由于在引导交换时 0080H 和 1080H 要相互切换，因此要对 0080H 设置与 1080H 相同的值。

注意事项
**1.** 禁止设置：**WDCS2 = WDCS1 = WDCS0 = 0** 且 **WINDOW1 = WINDOW0 = 0**
**2. Flash** 存储器自编程和模拟 **EEPROM** 期间看门狗定时器继续工作。处理过程中中断响应时间将被延迟。设置溢出时间和窗口大小时应考虑到这一延迟情况。
**3.** 如果 **LSROSC = 0(**可由软件停止振荡**)**，则无论内部振荡模式寄存器**(RCM)**的第 **0** 位**(LSRSTOP)**为何值，在 **HALT** 和 **STOP** 模式下对看门狗定时器不提供计数时钟。
当 **8** 位定时器 **H1** 使用内部低速振荡时钟时，即使在 **HALT/STOP** 模式下也对 **8** 位定时器 **H1** 提供计数时钟。
**4.** 必须对第 **7** 位清零**(0)**。

备注
**1.** $f_{RL}$: 内部低速振荡时钟频率
**2.** ( ): $f_{RL}$ = 264 kHz (最大)

图 **24-1.** 选项字节的格式 **(2/2)**

地址: 0081H/1081H[注 1, 2]

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | POCMODE |

| POCMODE | POC 模式选择 |
|---|---|
| 0 | 1.59 V POC 模式 (默认) |
| 1 | 2.7 V/1.59 V POC 模式 |

注　**1.** 使用专用 Flash 存储器编程器只能对 POCMODE 进行写操作。在自编程期间不能设置 POCMODE，或者自编程引导交换(此时，设置 1.59 V POC 模式(默认))操作时也不能设置。但由于引导交换时 1081H 单元的值被复制到 0081H 单元，建议在使用引导交换功能时为 1081H 单元设置与 0081H 相同的值。

**2.** 要改变 POC 模式的设置，可在对 Flash 存储器进行一次性擦除(片擦除)后将该值再次写入 0081H 中。在对 Flash 存储器指定 block 擦除后不能修改设置。

**注意事项　必须将第 7 ～ 1 位清零(0)。**

地址: 0082H/1082H, 0083H/1083H[注]

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

注　**由于 0082H 与 0083H 单元为保留区域，因此必须对这两个存储单元赋值 00H。由于在引导交换时 0082H 与 0083H 要和 1082H 与 1083H 相互切换，因此也要对 1082 与 1083H 设置 00H。**

地址: 0084H/1084H[注 1, 2]

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | OCDEN1 | OCDEN0 |

| OCDEN1 | OCDEN0 | 片上调试操作控制 |
|---|---|---|
| 0 | 0 | 禁止操作 |
| 0 | 1 | 禁止设置 |
| 1 | 0 | 允许操作。在确定片上调试安全 ID 失效时不要擦除 Flash 存储器数据 。 |
| 1 | 1 | 允许操作。在确定片上调试安全 ID 失效时可以擦除 Flash 存储器数据 。 |

注　**1.** 对于没有配备片上调试功能的产品(μPD78F0511, 78F0512, 78F0513, 78F0514, 和 78F0515)，必须将 00H 赋给 0084H 单元(禁止片上调试操作)。由于在引导交换时 0084H 和 1084H 要相互切换，因此也要对 1084H 设置 00H。

**2.** 对于具有片上调试功能的产品(μPD78F0513D, 78F0515D)，如果要使用它的片上调试功能，需要对 0084H 单元赋值 02H 或 03H。由于在引导交换时 0084H 和 1084H 要相互切换，因此也要对 1084H 设置与 0084H 相同的值。

**备注**　如需了解片上调试安全 ID 的详细信息，参见**第二十六章　片上调试功能 (仅μPD78F0513D, 78F0515D)**。

以下是设置选项字节的软件描述示例

```
OPT       CSEG   AT 0080H
OPTION:   DB     30H          ; 允许看门狗定时器的操作 (对非法访问检测的操作),
                              ; 看门狗定时器窗口打开周期: 50%,
                              ; 看门狗定时器的溢出时间: 2^10/fRL,
                              ; 可由软件停止内部低速振荡器
          DB     00H          ; 1.59 V POC 模式
          DB     00H          ; 保留区域
          DB     00H          ; 保留区域
          DB     00H          ; 禁止片上调试
```

**备注**　复位处理期间参考选项字节。如需了解复位处理的时序，可参见**第二十一章 复位功能**。

# 第二十五章　Flash 存储器

78K0/KC2 包含 Flash 存储器，可以安装在电路板上实现在线写入、擦除和覆盖操作。

## 25.1 内部存储器容量切换寄存器

可以使用内部存储器容量切换寄存器（IMS）选择内存容量。
可使用 8 位存储器操作指令设置 IMS。
复位信号的产生将 IMS 设置为 CFH。

**注意事项　复位释放后必须按照表 25-1 对每个产品进行设置。**

**图 25-1. 内部存储器容量切换寄存器 （IMS） 的格式**

地址: FFF0H　复位后: CFH　R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| IMS | RAM2 | RAM1 | RAM0 | 0 | ROM3 | ROM2 | ROM1 | ROM0 |

| RAM2 | RAM1 | RAM0 | 内部高速 RAM 容量选择 |
|---|---|---|---|
| 0 | 0 | 0 | 768 字节 |
| 1 | 1 | 0 | 1024 字节 |
| 其它 | | | 禁止设置 |

| ROM3 | ROM2 | ROM1 | ROM0 | 内部 ROM 容量选择 |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 16 KB |
| 0 | 1 | 1 | 0 | 24 KB |
| 1 | 0 | 0 | 0 | 32 KB |
| 1 | 1 | 0 | 0 | 48 KB |
| 1 | 1 | 1 | 1 | 60 KB |
| 其它 | | | | 禁止设置 |

表 **25-1.** 内部存储器容量切换寄存器的设置

| Flash 存储器版本 (78K0/KC2) | IMS 设置 |
|---|---|
| μPD78F0511 | 04H |
| μPD78F0512 | C6H |
| μPD78F0513, 78F0513D | C8H |
| μPD78F0514 | CCH |
| μPD78F0515, 78F0515D | CFH |

## 25.2 内部扩展 RAM 容量切换寄存器

使用内部扩展 RAM 容量切换寄存器(IXS)可以选择内部扩展 RAM 的容量。
可使用 8 位存储器操作指令设置 IXS。
复位信号的产生将 IXS 设置为 0CH。

**注意事项 在复位释放后必须如表 25-2 所示设置每种产品的值。**

图 **25-2.** 内部扩展 **RAM** 容量切换寄存器 **(IXS)** 的格式

地址: FFF4H 复位后: 0CH R/W

| 符号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| IXS | 0 | 0 | 0 | IXRAM4 | IXRAM3 | IXRAM2 | IXRAM1 | IXRAM0 |

| IXRAM4 | IXRAM3 | IXRAM2 | IXRAM1 | IXRAM0 | 内部扩展 RAM 的容量选择 |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 0 | 0 字节 |
| 0 | 1 | 0 | 1 | 0 | 1024 字节 |
| 0 | 1 | 0 | 0 | 0 | 2048 字节 |
| 其它 | | | | | 禁止设置 |

表 **25-2.** 内部扩展 **RAM** 容量切换寄存器 **(IXS)** 的设置

| Flash 存储器版本 (78K0/KC2) | IXS 设置 |
|---|---|
| μPD78F0511 | 0CH |
| μPD78F0512 | |
| μPD78F0513, 78F0513D | |
| μPD78F0514 | 0AH |
| μPD78F0515, 78F0515D | 08H |

## 25.3 用 Flash 编程器写入数据

可使用专用 Flash 编程器将数据写入 Flash 存储器，可以 On-board 写入，也可以 Off-board 写入。

**(1) On-board 编程**

当目标系统上安装了 78K0/KC2 系列产品后，Flash 存储器的内容可以被重写。必须将连接专用 Flash 存储器编程器的连接器安装在目标系统中。

**(2) Off-board 编程**

在将 78K0/KC2 系列产品安装到目标系统之前，可使用专用程序适配器（FA 系列）将数据写入 Flash 存储器。

备注　FA 系列是 Naito Densei Machida Mfg. Co., Ltd.的产品。

**表 25-3. 78K0/KC2** 系列产品与专用 **Flash** 编程器之间的写操作

**(a) 44** 引脚产品

| 专用 Flash 编程器的引脚配置 | | | 使用 CSI10 | | 使用 UART6 | |
|---|---|---|---|---|---|---|
| 信号名称 | I/O | 引脚功能 | 引脚名称 | 引脚编号 | 引脚名称 | 引脚编号 |
| SI/RxD | 输入 | 接收信号 | SO10/P12 | 29 | TxD6/P13 | 28 |
| SO/TxD | 输出 | 发送信号 | SI10/RxD0/P11 | 30 | RxD6/P14 | 27 |
| SCK | 输出 | 传输时钟 | $\overline{\text{SCK10}}$/TxD0/P10 | 31 | – | – |
| CLK | 输出 | 78K0/KC2 的时钟 | –[注 1] | – | **注 2** | **注 2** |
| /RESET | 输出 | 复位信号 | $\overline{\text{RESET}}$ | 3 | $\overline{\text{RESET}}$ | 3 |
| FLMD0 | 输出 | 模式信号 | FLMD0 | 6 | FLMD0 | 6 |
| $V_{DD}$ | I/O | $V_{DD}$ 电压产生/电压监测 | $V_{DD}$ | 11 | $V_{DD}$ | 11 |
| | | | $AV_{REF}$ | 32 | $AV_{REF}$ | 32 |
| GND | – | 地 | $V_{SS}$ | 10 | $V_{SS}$ | 10 |
| | | | $AV_{SS}$ | 33 | $AV_{SS}$ | 33 |

注 **1.** 在使用 CSI10 时，只能使用内部高速振荡时钟 ($f_{RH}$)。

<R> **2.** 在使用 UART6 时，只能使用 X1 时钟 ($f_X$) 或外部主系统时钟 ($f_{EXCLK}$)。若要使用 Flash 编程器的时钟，则根据不同的编程器有不同的连接方法。

- PG-FP4, FL-PR4: 连接编程器的 CLK 到 EXCLK/X2/P122 (引脚 7)。
- PG-FPL3, FP-LITE3: 连接编程器的 CLK 到 X1/P121 (引脚 8)，并连接其反向信号到 X2/EXCLK/P122 (引脚 7)。

**(b) 48** 引脚产品

| 专用 Flash 编程器的引脚配置 | | | 使用 CSI10 | | 使用 UART6 | |
|---|---|---|---|---|---|---|
| 信号名称 | I/O | 引脚功能 | 引脚名称 | 引脚编号 | 引脚名称 | 引脚编号 |
| SI/RxD | 输入 | 接收信号 | SO10/P12 | 20 | TxD6/P13 | 19 |
| SO/TxD | 输出 | 发送信号 | SI10/RxD0/P11 | 21 | RxD6/P14 | 18 |
| SCK | 输出 | 传输时钟 | $\overline{\text{SCK10}}$/TxD0/P10 | 22 | – | – |
| CLK | 输出 | 78K0/KC2 的时钟 | –[注 1] | – | **注 2** | **注 2** |
| /RESET | 输出 | 复位信号 | $\overline{\text{RESET}}$ | 40 | $\overline{\text{RESET}}$ | 40 |
| FLMD0 | 输出 | 模式信号 | FLMD0 | 43 | FLMD0 | 43 |
| $V_{DD}$ | I/O | $V_{DD}$ 电压产生/电压监测 | $V_{DD}$ | 48 | $V_{DD}$ | 48 |
| | | | $AV_{REF}$ | 23 | $AV_{REF}$ | 23 |
| GND | – | 地 | $V_{SS}$ | 47 | $V_{SS}$ | 47 |
| | | | $AV_{SS}$ | 24 | $AV_{SS}$ | 24 |

注 **1.** 在使用 CSI10 时，只能使用内部高速振荡时钟 ($f_{RH}$)。

<R> **2.** 在使用 UART6 时，只能使用 X1 时钟 ($f_X$) 或外部主系统时钟 ($f_{EXCLK}$)。若要使用 Flash 编程器的时钟，则根据不同的编程器有不同的连接方法。

- PG-FP4, FL-PR4: 连接编程器的 CLK 到 EXCLK/X2/P122 (引脚 44)。
- PG-FPL3, FP-LITE3: 连接编程器的 CLK 到 X1/P121 (引脚 45)，并连接其反向信号到 X2/EXCLK/P122 (引脚 44)。

在使用适配器对 Flash 存储器进行写操作时可采用如下所示的连接方式。

**图 25-3. 3 线串行 I/O (CSI10) 模式下使用适配器对 Flash 存储器进行写操作示例(44 引脚产品)**

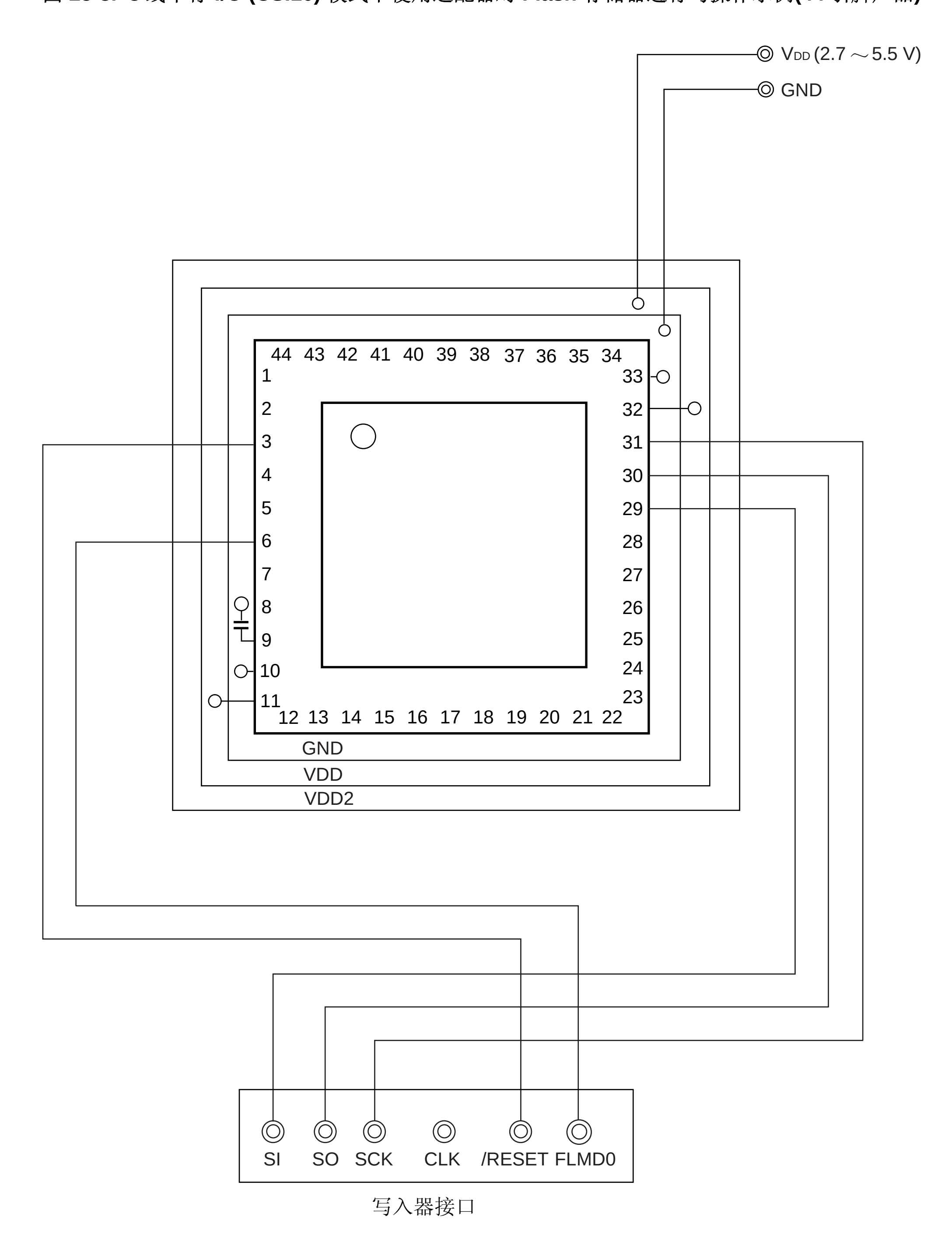
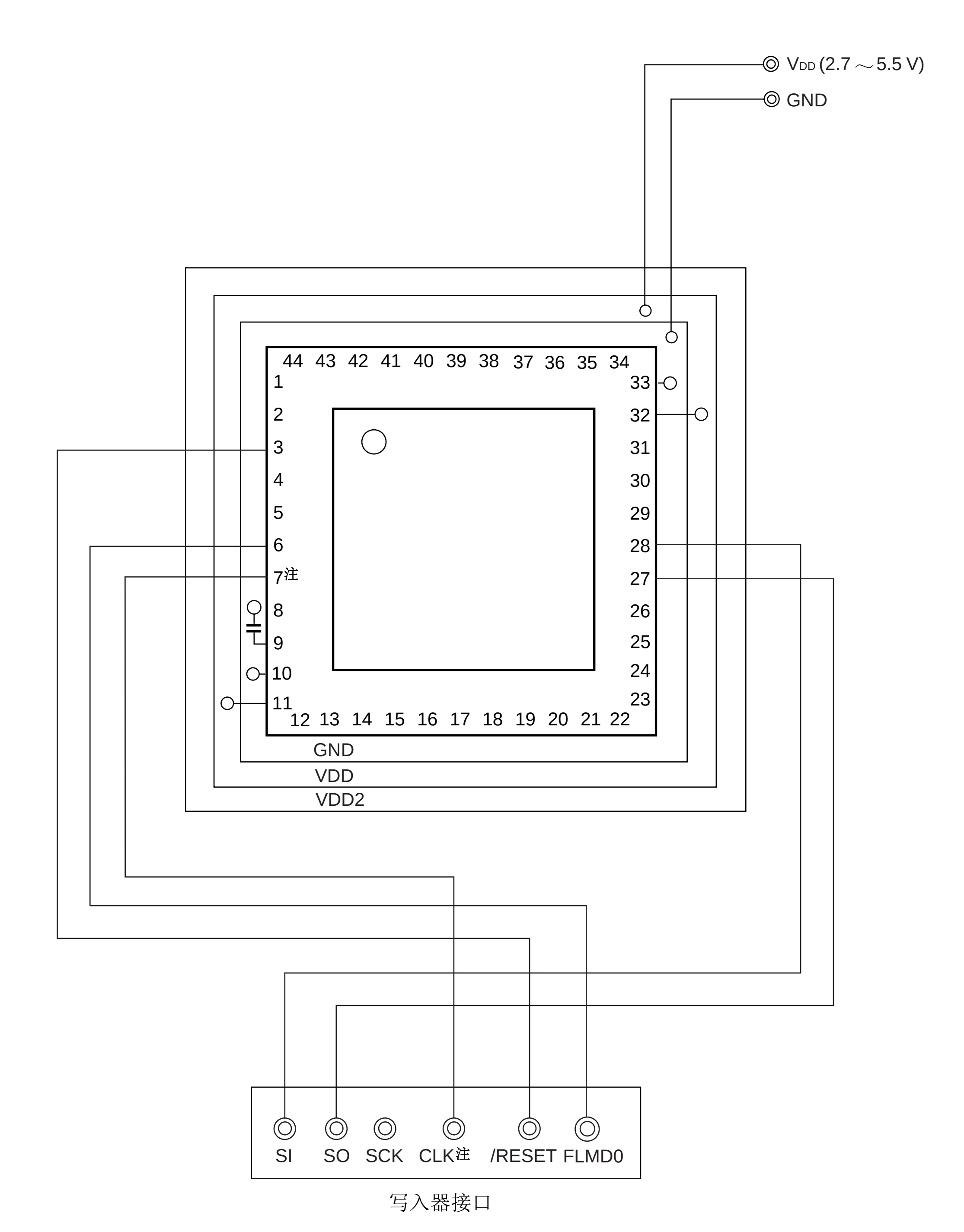


图 25-4. UART (UART6) 模式下使用适配器对 Flash 存储器进行写操作示例(44 引脚产品)




<R>　注　上图举例当使用 PG-FP4 或 FL-PR4 的时钟输出时的连线。
当使用 PG-FPL3 或 FP-LITE3 的时钟输出时，连接 CLK 到 X1/P121 (引脚 8)，并且连接它的反向信号到 X2/EXCLK/P122 (引脚 7)。

**图 25-5. 3 线串行 I/O (CSI10) 模式下使用适配器对 Flash 存储器进行写操作示例(48 引脚产品)**

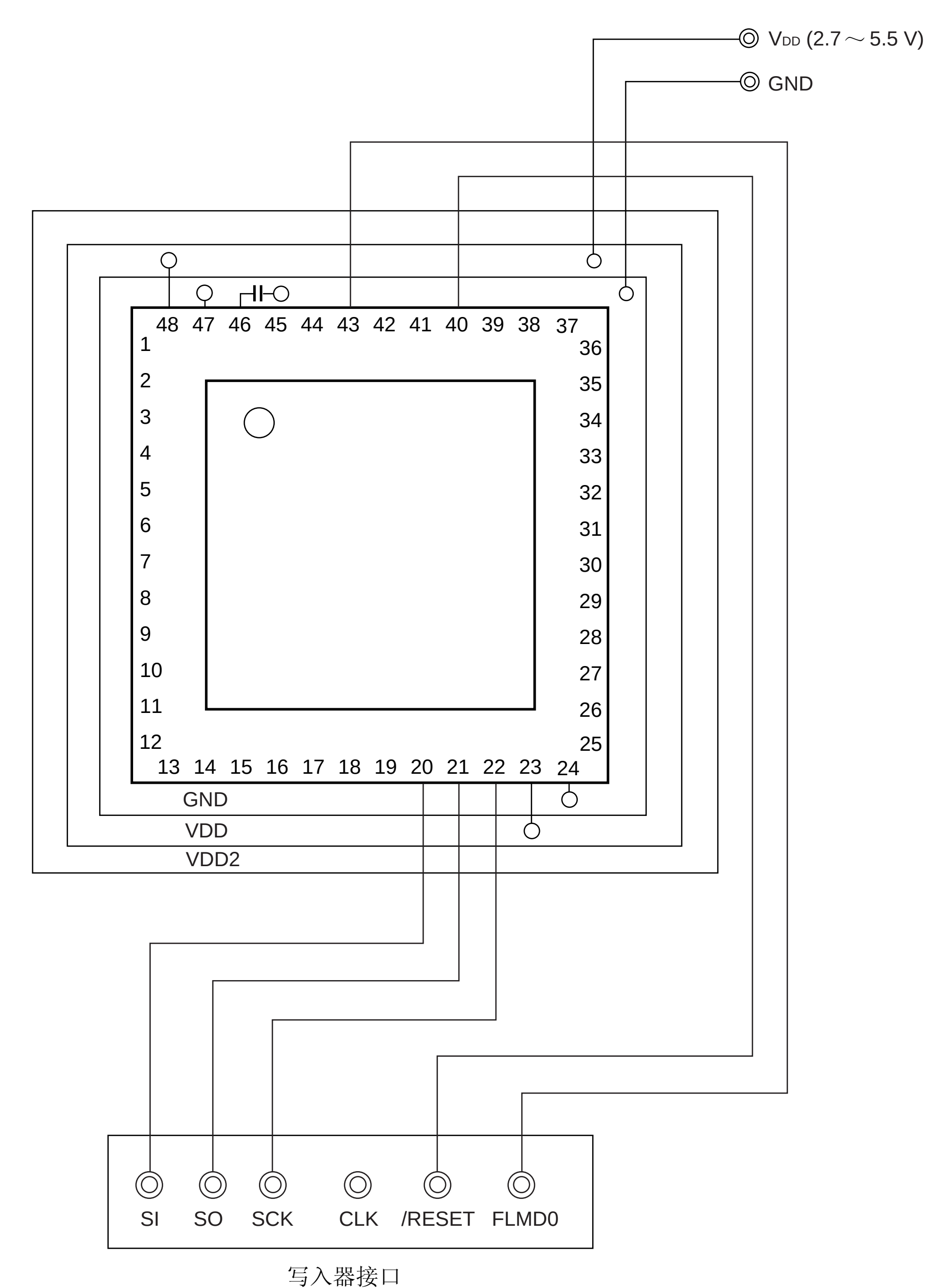
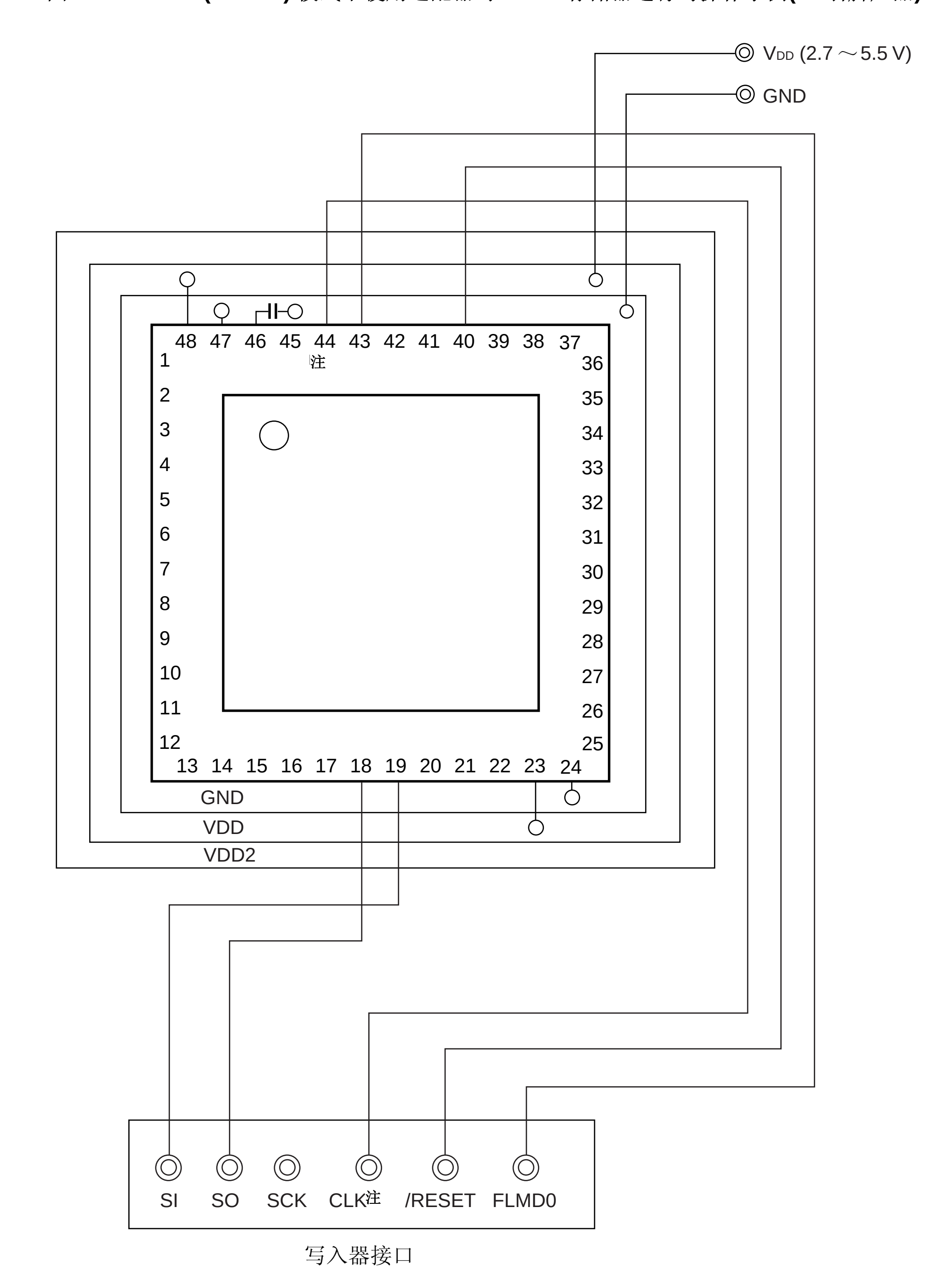


图 **25-6. UART (UART6)** 模式下使用适配器对 **Flash** 存储器进行写操作示例**(48** 引脚产品**)**

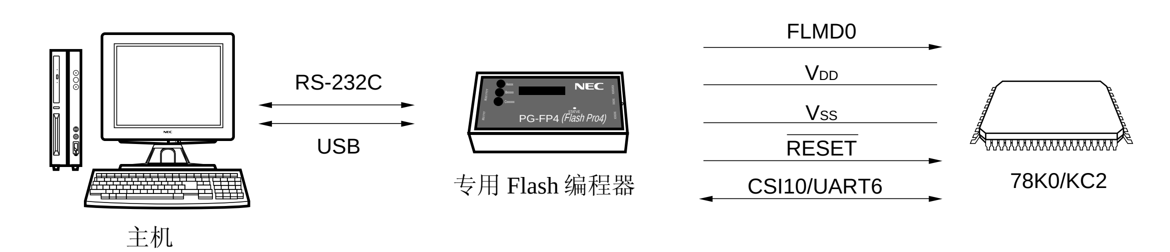



<R>　注　上图举例当使用 PG-FP4 或 FL-PR4 的时钟输出时的连线。
当使用 PG-FPL3 或 FP-LITE3 的时钟输出时，连接 CLK 到 X1/P121 (引脚 45)，并且连接它的反向信号到 X2/EXCLK/P122 (引脚 44)。

## 25.4 编程环境

以下是 78K0/KC2 Flash 存储器所需的编程环境。

**图 25-7. Flash 存储器编程环境**




需要有一个控制专用 Flash 编程器的主机。

专用 Flash 编程器与 78K0/KC2 的接口使用 CSI10 或 UART6 进行写和擦除等操作。Flash 存储器离线写操作时必须使用一个专用程序适配器（FA 系列）。

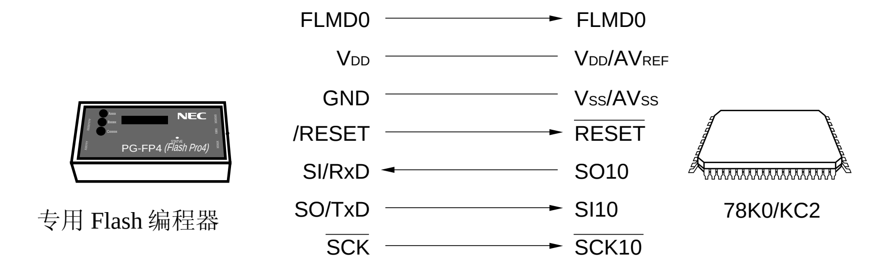
## 25.5 通信模式

通过 78K0/KC2 的 CSI10 或 UART6，在专用 Flash 存储器编程器与 78K0/KC2 之间建立串行通信。

**(1) CSI10**

传输速率: 2.4 kHz ～ 2.5 MHz

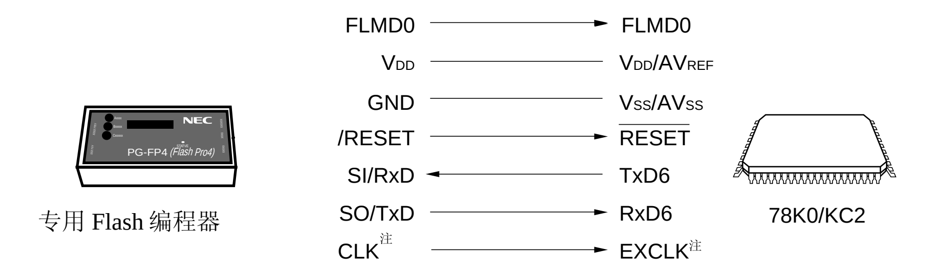
**图 25-8. 专用 Flash 存储器编程器的通信 (CSI10)**



**(2) UART6**

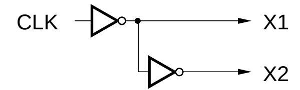
传输速率: 115200 bps

**图 25-9. 专用 Flash 编程器的通信(UART6)**




<R> 注 以上举例为当使用 PG-FP4 或 FL-PR4 的时钟输出的连线。

当使用 PG-FPL3 或 FP-LITE3 的时钟输出时，连接 CLK 到 X1/P121，并且连接它的反转信号到 X2/EXCLK/P122 。




对于 78K0/KC2 专用 flash 存储编程器产生以下信号。详细情况，请参看 PG-FP4, FL-PR4, PG-FPL3, 或 FP-LITE3 的用户手册。

**表 25-4. 引脚连接**

| 专用 Flash 存储器编程器 | | | 78K0/KC2 | 连接 | |
|---|---|---|---|---|---|
| 信号名 | I/O | 引脚功能 | 引脚名 | CSI10 | UART6 |
| FLMD0 | 输出 | 模式信号 | FLMD0 | ◎ | ◎ |
| $V_{DD}$ | I/O | $V_{DD}$ 电压/电源监测 | $V_{DD}$, $AV_{REF}$ | ◎ | ◎ |
| GND | – | 地 | $V_{SS}$, $AV_{SS}$ | ◎ | ◎ |
| CLK | 输出 | 时钟输出到 78K0/KC2 | 注 1 | ×注 2 | ○注 1 |
| /RESET | 输出 | 复位信号 | $\overline{\text{RESET}}$ | ◎ | ◎ |
| SI/RxD | 输入 | 接收信号 | SO10/TxD6 | ◎ | ◎ |
| SO/TxD | 输出 | 发送信号 | SI10/RxD6 | ◎ | ◎ |
| SCK | 输出 | 传输时钟 | $\overline{\text{SCK10}}$ | ◎ | × |

<R> 注

1. 当使用 UART6 时，只能使用 X1 时钟 ($f_X$) 或外部主系统时钟 ($f_{EXCLK}$) 。当使用专用 flash 存储器编程器的时钟输出时，依据专用 flash 存储器编程器使用类型改变引脚连接。
   - PG-FP4, FL-PR4: 连接编程器的 CLK 到 EXCLK/X2/P122 。
   - PG-FPL3, FP-LITE3: 连接编程器的 CLK 到 X1/P121，并且连接它的反向信号到 X2/EXCLK/P122 。
2. 当使用 CSI10 时，只能使用内部高速振荡时钟 ($f_{RH}$) 。

备注 ◎: 确保连接引脚。

○: 如果在目标板上产生信号，则此引脚不需要连接。

×: 此引脚不需要连接。

## 25.6 在线方式的引脚连接

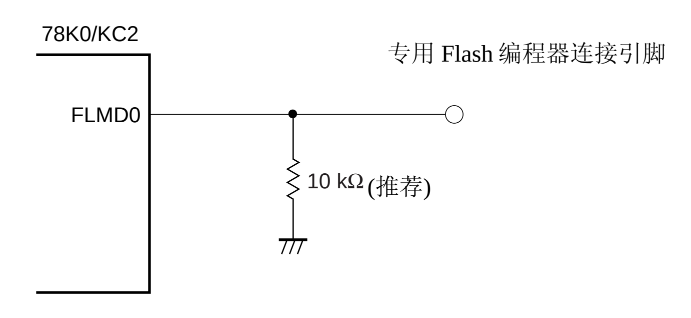
对 Flash 存储器进行在线写操作时，目标系统必须有连接专用 Flash 存储器编程器的连接器。电路板上首先要提供一个选择功能，可以选择正常操作模式或 Flash 存储器编程模式。

当设置 Flash 存储器编程模式时，那些不用于 Flash 存储器编程的引脚状态与复位后的状态相同。因此如果外部设备不能立即识别复位后的状态，则必须采用如下方式连接引脚。

### 25.6.1 FLMD0 引脚

在普通操作模式下，FLMD0 引脚的输入电压为 0V 。在 Flash 存储器编程模式中，将 $V_{DD}$ 写电压提供给 FLMD0 引脚。FLMD0 引脚的连接示意图如下所示。

<R> **图 25-10. FLMD0 引脚连接示例**




### 25.6.2 串行接口引脚

串行接口使用的引脚如下所示。

**表 25-5. 串行接口使用的引脚**

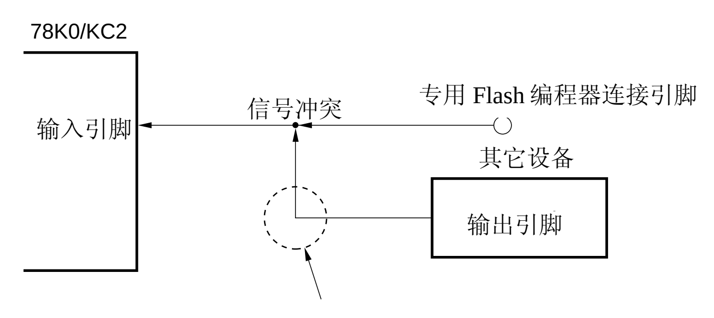
| 串行接口 | 使用的引脚 |
|---|---|
| CSI10 | SO10, SI10, $\overline{\text{SCK10}}$ |
| UART6 | TxD6, RxD6 |

在将专用 Flash 存储器编程器和串行接口 (已与板上其它设备连接) 的引脚相连时，必须特别注意：信号之间不能冲突，并且另一个设备不能出现误动作。

**(1) 信号冲突**

如果专用 Flash 编程器 (输出端) 与串行接口 (已连接到另一个设备的输出端) 的一个输入引脚相连，这时会产生信号冲突。为了避免这种情况，应隔离与另一个设备的连接，或者使另一个设备处于高阻抗状态。

**图 25-11. 信号冲突 （串行接口的输入引脚）**

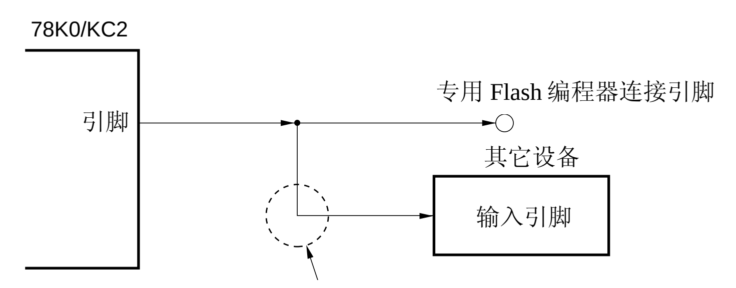



在 Flash 存储器编程模式下，设备的输出信号与专用 Flash 编程器发送的信号发生冲突。因此必须隔离该设备的信号。

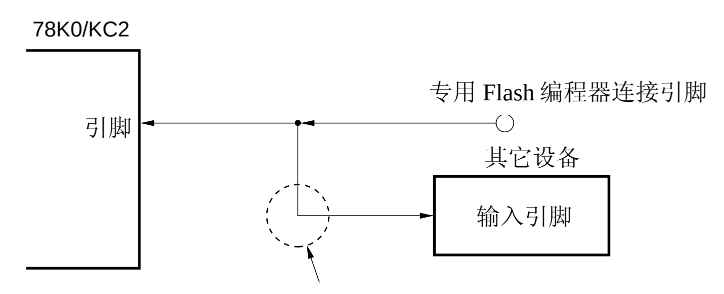
**(2) 其它设备的故障**

如果专用 Flash 存储器编程器的输出或输入端与串行接口（已连接到另一个设备的输入端）的一个输入或输出引脚相连，则信号可能会输出到另一个设备，从而引起该设备的故障。为了避免这种情况，应隔离与该设备的连接。

**图 25-12. 其它设备的故障**




在 Flash 存储器编程模式下，如果 78K0/KC2 的输出信号影响到另外一个设备,那么隔离另一设备的信号



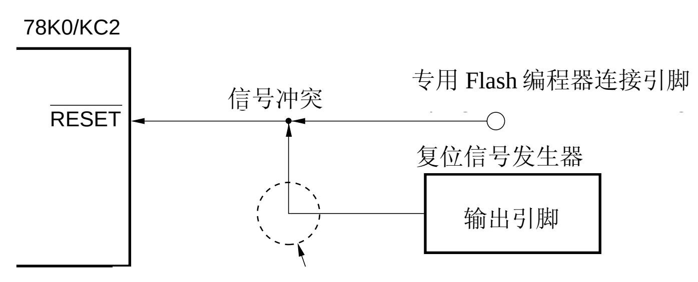

在 Flash 存储器编程模式下，如果专用 Flash 编程器的输出信号影响到另外一个设备,那么隔离另一设备的信号

### 25.6.3 $\overline{\text{RESET}}$ 引脚

如果将专用 Flash 编程器的复位信号连接到 $\overline{\text{RESET}}$ 引脚（已连接到板上的复位信号发生器），则会产生信号冲突。为了避免这种情况，应隔离与复位信号发生器的连接。

在 Flash 存储器编程模式下，如果从用户系统输入复位信号，则不能对 Flash 存储器进行正确编程。因此除了专用 Flash 存储器编程器的复位信号外，不要输入其它信号。

**图 25-13. 信号冲突 （$\overline{\text{RESET}}$ 引脚）**




在 Flash 存储器编程模式中，复位信号发生器输出的信号与专用 Flash 存储器编程器的输出信号发生冲突。因此必须隔离复位信号发生器的信号。

### 25.6.4 端口引脚

当设置 Flash 存储器编程模式时，那些不用于 Flash 存储器编程的引脚状态与复位后的状态相同。因此，如果外部设备(与端口相连)不能立即识别复位后的状态，则必须通过一个电阻将端口引脚连接到 $V_{DD}$ 或 $V_{SS}$。

### 25.6.5 REGC 引脚

以普通操作相同的方式连接 REGC 引脚到 GND 通过一个电容 (0.47 ～ 1 $\mu$F：推荐) 。

### 25.6.6 其它信号引脚

在普通操作模式下使用 on-board 时钟时，以相同的状态连接 X1 和 X2 。

<R> 为了从专用 flash 存储器编程器输入操作时钟，要按以下进行连接。

- PG-FP4, FL-PR4: 连接编程器的 CLK 到 EXCLK/X2/P122 。
- PG-FPL3, FP-LITE3: 连接编程器的 CLK 和 X1/P121，并且连接它的反向信号到 X2/EXCLK/P122 。

注意事项 **1. 在使用 CSI10 时，只能使用内部高速振荡时钟 ($f_{RH}$)。**

**2. 在使用 UART6 时，只能使用 X1 时钟 ($f_X$)或外部主系统时钟 ($f_{EXCLK}$)。**

<R> 注意事项 **3.** 对于具有 **48KB**或更大**Flash**存储器而没有片上调试功能的产品 **(**$\mu$**PD78F0514** 和 **78F0515)**，有一个 **“I”, “K”,** 或 **“E”** 的产品等级，对于具有片上调试功能的产品 **(**$\mu$**PD78F0513D** 和 **78F0515D)**，在使用 **Flash**编程器写**Flash**存储器时，按以下连接**P31/INTP2/OCD1A**[注] 和 **P121/X1/OCD0A**[注]。

- **P31/INTP2/OCD1A**[注]**:** 通过电阻连接到 $V_{SS}$ **(10 kΩ:** 推荐**)**。
- **P121/X1/OCD0A**[注]**:** 当使用此引脚作为一个端口时，通过电阻连接到 $V_{SS}$ **(10 kΩ:** 推荐**)(**在输入模式**)**或悬空**(**在输出模式**)**。

当通过自编程方式写入 **Flash** 存储器时，以上连接并不需要。

<R> 注 只有 $\mu$PD78F0513D 和 78F0515D 提供 OCD0A 和 OCD1A 。

<R> 备注 对于产品级别，请咨询 NEC 电子的销售代表。

### 26.6.7 电源

要使用 Flash 存储器编程器输出的电源电压，则将 $V_{DD}$ 引脚与 Flash 存储器编程器的 $V_{DD}$ 相连，将 $V_{SS}$ 引脚与 Flash 存储器编程器的 GND 相连。

然而，如果要使用 Flash 存储器编程器的电压监视功能，必须将 $V_{DD}$ 和 $V_{SS}$引脚分别与 Flash 存储器编程器的 $V_{DD}$
<R> 和 GND 相连，即使是 on-board 电源电压。

其它的供电电压（$AV_{REF}$ 和 $AV_{SS}$）与正常操作模式中的相同。

## 25.7 编程方法

### 25.7.1 控制 Flash 存储器

下图显示了 Flash 存储器操作过程。

**图 25-14. Flash 存储器操作过程**

开　始

提供 FLMD0 脉冲 → 设置 Flash 存储器编程模式

选择通信模式

对 Flash 存储器进行操作

结束?

否

是

结束

### 25.7.2 Flash 存储器编程模式

如果通过专用 Flash 编程器重写 Flash 存储器的内容，则必须将 78K0/KC2 设置为 Flash 存储器编程模式。而要设置该模式，必须将 FLMD0 引脚连接到 $V_{DD}$，并对复位信号清零。

当在线写 Flash 存储器时，使用跳线改变模式。

**图 25-15. Flash 存储器编程模式**

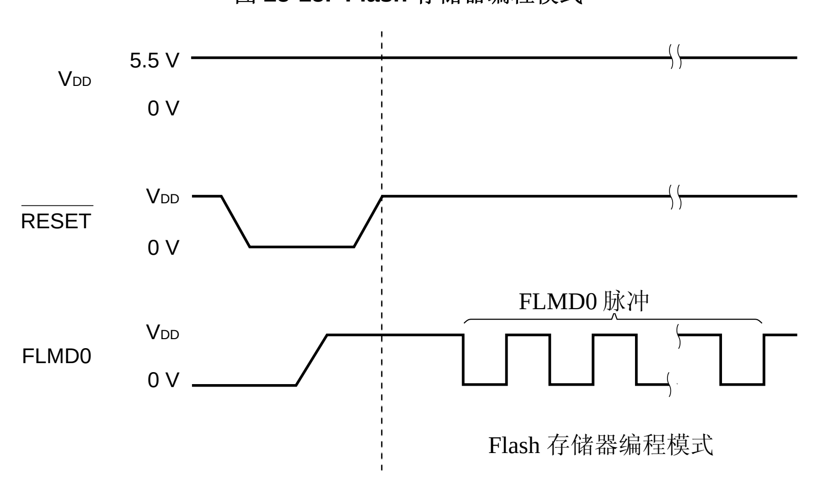



**表 25-6. FLMD0 引脚与复位释放后操作模式之间的关系**

| FLMD0 | 操作模式 |
|---|---|
| 0 | 正常操作模式 |
| $V_{DD}$ | Flash 存储器编程模式 |

### 25.7.3 选择通信模式

在 78K0/KC2 中，进入专用 Flash 存储器编程模式后，通过将脉冲（可多达 11 个脉冲）输入到 FLMD0 引脚，来选择通信模式。由 Flash 存储器编程器产生这些 FLMD0 脉冲。

下表显示了脉冲个数与通信模式之间的关系。

<R> **表 25-7. 通信模式**

| 通信模式 | 标准设置[注 1] | | | | 使用的引脚 | 外部时钟 | FLMD0 脉冲数 |
|---|---|---|---|---|---|---|---|
| | 端口 | 速率 | 频率 | 倍率 | | | |
| UART (UART6) | UART-Ext-Osc | 115200 bps[注 3] | 2 ～ 20 MHz[注 2] | 1.0 | TxD6, RxD6 | $f_X$ | 0 |
| | UART-Ext-FP4CK | | | | | $f_{EXCLK}$ | 3 |
| 3 线串行 I/O (CSI10) | CSI-Internal-OSC | 2.4 kHz ～ 2.5 MHz | – | | SO10, SI10, SCK10 | $f_{RH}$ | 8 |

注 **1.** 在 Flash 存储编程器的用户图形界面应选择标准设置项目。

**2.** 电压不同设置范围也不一样。如需了解详细内容，可参见电气特性章节。

**3.** 波特率误差以外的因素，如信号波形瞬变，也会影响 UART 通信，因此必须象测量波特率误差一样严格测量瞬变。

**注意事项 当选择 UART6 时，在接收到 FLMD0 脉冲后，根据专用 Flash 存储器编程器发送的复位命令计算接收时钟。**

备注 $f_X$: X1 时钟

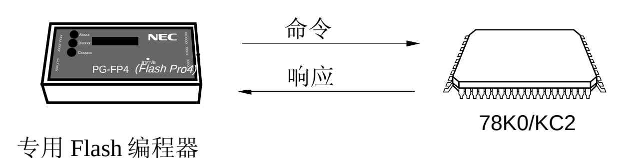
$f_{EXCLK}$: 外部主系统时钟

$f_{RH}$: 内部高速振荡时钟

### 25.7.4 通信命令

78K0/KC2 使用命令与专用 Flash 存储器编程器进行通信。从 Flash 编程器发往 78K0/KC2 的信号称为命令，从 78K0/KC2 发往专用 Flash 存储器编程器的命令称为响应命令。

**图 25-16. 通信命令**




78K0/KC2 的 Flash 存储器控制命令列表如下。所有这些命令都从编程器发出，78K0/KC2 根据相应的命令进行处理。

<R>

**表 25-8. Flash 存储器控制命令**

| 类别 | 命令名称 | 功能 |
|---|---|---|
| 校验 | 校验 | 比较 flash 存储器指定区域的内容和编程器中数据发送的内容。 |
| 擦除 | 片擦除 | 擦除整个 Flash 存储器 |
| | Block 擦除 | 擦除 Flash 存储器的指定区域 |
| 空白检测 | Block 空白检测 | 验证 Flash 存储器的指定 block 是否被正确擦除 |
| 写 | 编程 | 向 Flash 存储器的指定区域写入数据 |
| 取得信息 | 状态 | 取得当前操作状态 (状态数据) |
| | 硅标记命令 | 取得 78K0/Kx2 信息 (例如编号和 Flash 存储器配置) |
| | 取得版本 | 取得 78K0/Kx2 版本和固件版本 |
| | 校验和 | 从指定区域取得校验和数据 |
| 安全 | 安全设定 | 设定安全信息 |
| 其它 | 复位 | 用于检测通讯同步状态 |
| | 振荡频率设定 | 指定振荡频率 |

78K0/KC2 根据专用 Flash 编程器发出的命令返回一个响应命令。78K0/KC2 发出的响应命令列表如下。

**表 25-9. 响应命令**

| 命令名称 | 功能 |
|---|---|
| ACK | 响应命令/数据 |
| NAK | 响应非法命令/数据 |

## 25.8 安全设置

<R> 78K0/KC2 支持一个安全功能，其禁止重写写入内部 flash 存储器的用户程序，所以未被授权的人不能改变程序。
以下显示使用安全设定命令能够执行的操作。在设置下一个编程模式时，安全性设置有效。

<R> • 禁止一次性擦除 (片擦除)
该项设置禁止在 on-board/off-board 编程时对 Flash 存储器所有 block 进行擦除和一次性擦除(片擦除)。一旦禁止执行一次性擦除(片擦除)命令，所有禁止设置项(包括一次性擦除(片擦除))不能被取消。

**注意事项 在对一次性擦除进行安全性设置后，不能再对该设置执行擦除操作。此外，由于禁止执行擦除命令，即使执行了写命令，与已经写到 Flash 存储器中的数据不同的数据也不能被写入。**

<R> • 禁止 block 擦除
该项设置禁止在 on-board/off-board 编程时对 Flash 存储器中的 block 进行擦除。而在使用自编程时 block 可以被擦除。

<R> • 禁止写
该项设置禁止在 on-board/off-board 编程时对 Flash 存储器所有 block 进行写和擦除。而在使用自编程时 block 可以被写入。

• 禁止重写引导簇 0
该项设置禁止对 Flash 存储器中引导簇 0(0000H ～ 0FFFH)进行一次性擦除(片擦除)、块擦除和写操作。

**注意事项 如果对重写引导簇 0 进行了安全性设置，则不能再对该设备的引导簇 0 进行重写。**

<R> 当 flash 存储器装载时通过缺省设置打开一次性擦除(片擦除)、block 擦除、写和重写引导簇 0 功能。通过 on-board/off-board 编程和自编程可以设置安全设定。且安全设定可以相互结合使用。

Flash 存储器在默认设置下，允许使用一次性擦除(片擦除)、块擦除、写和重写引导簇 0 命令。以上安全性设置仅用于在线/离线编程。各项安全性设置可以结合使用。

通过执行一次性擦除 (片擦除)命令，block 擦除和写入的禁止项被清零。

表 25-10 显示了当允许使用 78K0/KC2 安全性功能时擦除命令与写命令之间的关系。

<R>

**表 25-10. 当允许使用安全功能时命令之间的关系**

**(1) 在 on-board/off-board 编程时**

| 安全项 | 执行命令 | | |
|---|---|---|---|
| | 一次性擦除(片擦除) | Block 擦除 | 写入 |
| 禁止一次性擦除(片擦除) | 不能被一次性擦除. | Block 不能被擦除 | 可以执行[注] |
| 禁止 block 擦除 | 能被一次性擦除. | | 可以执行 |
| 禁止写入 | | | 不能执行 |
| 禁止重写引导簇 0 | 不能被一次性擦除. | 引导簇 0 不能被擦除 | 不能写入引导簇 0 |

注　由于禁止使用擦除命令，所以与已经写到 Flash 存储器中的不同的数据不能被写入。

**(2) 在自编程时**

| 安全项 | 执行命令 | |
|---|---|---|
| | Block 擦除 | 写入 |
| 禁止一次性擦除(片擦除) | Block 可以被擦除 | 可以执行 |
| 禁止 block 擦除 | | |
| 禁止写入 | | |
| 禁止重写引导簇 0 | 不能擦除引导簇 0 | 不能写入引导簇 0 |

如下表 25-11 所示在每个编程模式下怎样去执行安全设置。

<R>

**表 25-11. 每个编程模式下设定安全设置**

**(1) On-board/off-board 编程**

| 安全 | 安全设定 | 怎样取消安全设定 |
|---|---|---|
| 禁止一次性擦除(片擦除) | 通过专用 flash 存储编程器的用户界面等设定 | 设定后不能消除 |
| 禁止 block 擦除 | | 执行一次性擦除(片擦除)命令 |
| 禁止写入 | | |
| 禁止重写引导簇 0 | | 设定后不能消除 |

**(2) 自编程**

| 安全 | 安全设定 | 怎样取消安全设定 |
|---|---|---|
| 禁止一次性擦除(片擦除) | 通过信息库设定 | 设定后不能消除 |
| 禁止 block 擦除 | | 当 on-board/off-board 编程时，执行一次性擦除(片擦除)命令 (设定后不能消除) 。 |
| 禁止写入 | | |
| 禁止重写引导簇 0 | | 设定后不能消除 |

## <R> 25.9 当使用 PG-FP4 时，每个命令的处理时间 (推荐)

如下所示当使用 PG-FP4 作为专用 flash 存储编程器时，每个命令的处理时间 (推荐)。

**表 25-12. 当使用 PG-FP4 时每个命令的处理时间 (推荐)**

**(1) μPD78F0515, 78F0515D (内部 ROM 容量: 60 KB)**

| PG-FP4 的命令 | 端口： CSI-内部-OSC (内部高速振荡时钟 ($f_{RH}$)), 速度: 2.5 MHz | 端口: UART-外部-FP4CK (外部主系统时钟 ($f_{EXCLK}$)), 速度: 115,200 bps | |
|---|---|---|---|
| | | 频率: 2.0 MHz | 频率: 20 MHz |
| 信号 | 0.5 s (TYP.) | 0.5 s (TYP.) | 0.5 s (TYP.) |
| 空白验证 | 1 s (TYP.) | 1 s (TYP.) | 1 s (TYP.) |
| 擦除 | 1.5 s (TYP.) | 1 s (TYP.) | 1 s (TYP.) |
| 编程 | 5 s (TYP.) | 9 s (TYP.) | 9 s (TYP.) |
| 校验 | 2 s (TYP.) | 6.5 s (TYP.) | 6.5 s (TYP.) |
| E.P.V | 6 s (TYP.) | 10.5 s (TYP.) | 10.5 s (TYP.) |
| 校验和 | 0.5 s (TYP.) | 1 s (TYP.) | 1 s (TYP.) |
| 安全 | 0.5 s (TYP.) | 0.5 s (TYP.) | 0.5 s (TYP.) |

**(2) μPD78F0513, 78F0513D (内部 ROM 容量: 32 KB)**

| PG-FP4 的命令 | 端口： CSI-内部-OSC (内部高速振荡时钟 ($f_{RH}$)), 速度: 2.5 MHz | 端口: UART-外部-FP4CK (外部主系统时钟 ($f_{EXCLK}$)), 速度: 115,200 bps | |
|---|---|---|---|
| | | 频率: 2.0 MHz | 频率: 20 MHz |
| 信号 | 0.5 s (TYP.) | 0.5 s (TYP.) | 0.5 s (TYP.) |
| 空白验证 | 0.5 s (TYP.) | 0.5 s (TYP.) | 0.5 s (TYP.) |
| 擦除 | 0.5 s (TYP.) | 0.5 s (TYP.) | 0.5 s (TYP.) |
| 编程 | 2.5 s (TYP.) | 5 s (TYP.) | 5 s (TYP.) |
| 校验 | 1.5 s (TYP.) | 4 s (TYP.) | 3.5 s (TYP.) |
| E.P.V | 3.5 s (TYP.) | 6 s (TYP.) | 6 s (TYP.) |
| 校验和 | 0.5 s (TYP.) | 0.5 s (TYP.) | 0.5 s (TYP.) |
| 安全 | 0.5 s (TYP.) | 0.5 s (TYP.) | 0.5 s (TYP.) |

## 25.10 通过自编程进行 Flash 存储器编程

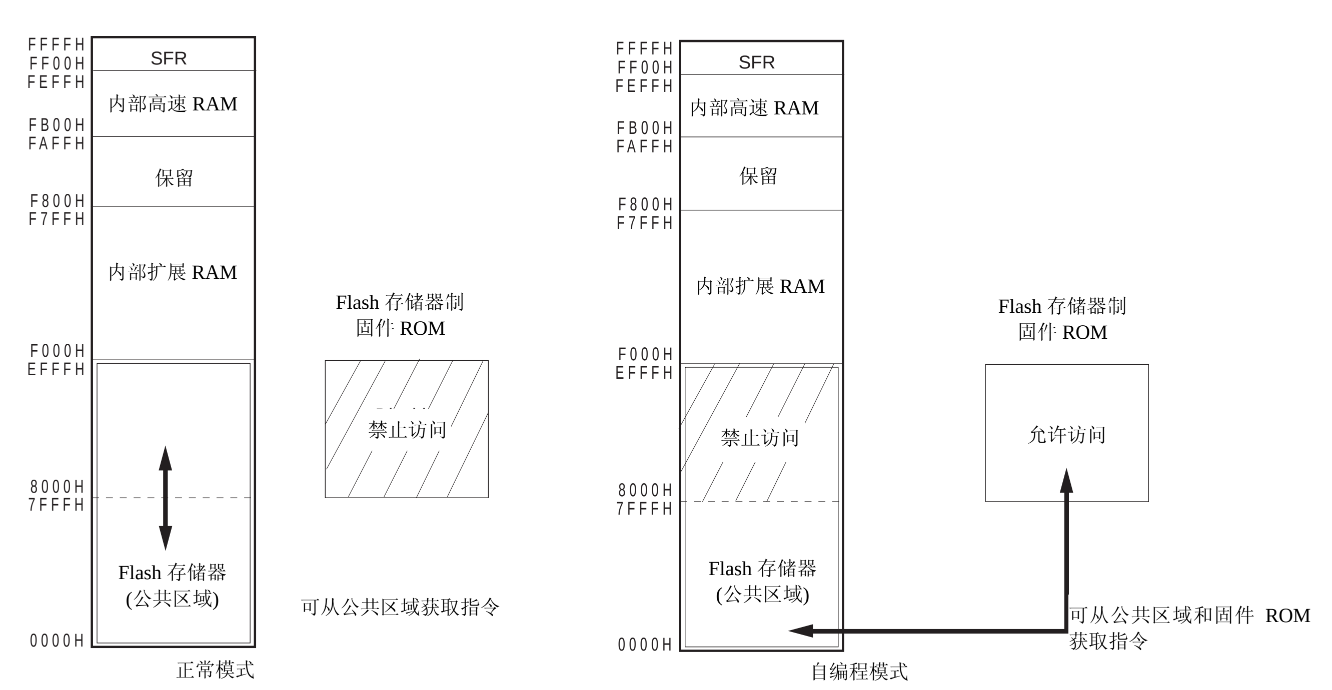
78K0/KC2 支持自编程功能，即可以通过用户程序重写 Flash 存储器。由于该功能允许用户应用程序使用 78K0/KC2 自编程库重写 Flash 存储器，因此可用于对程序升级。

如果自编程期间产生中断，可以暂时停止自编程操作，而去执行中断服务程序。要执行中断服务程序，应在停止自编程操作后恢复正常操作模式，并执行 EI 指令。而在恢复自编程模式后，可以恢复自编程操作。

备注　如需了解自编程功能和 78K0/KC2 自编程库的详细信息，可参看 **78K0/Kx2 Flash 存储器自编程用户手册 (U17516E)**。

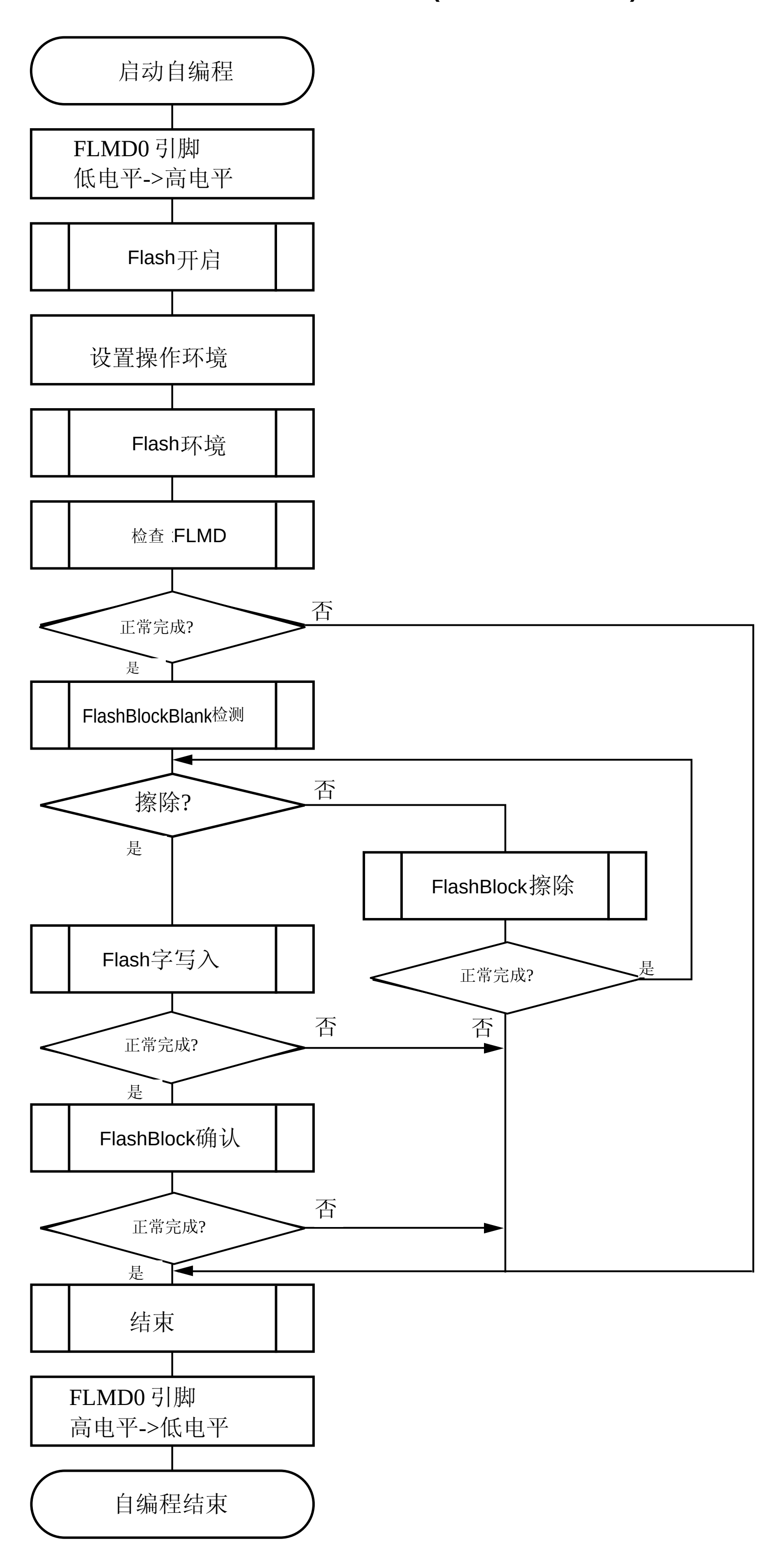
注意事项
**1.** 当 **CPU** 使用副系统时钟时不能使用自编程功能。
**2.** 自编程期间 **FLMD0** 引脚应输入高电平。
**3.** 在启动自编程之前必须执行 **DI** 指令。
自编程功能检查中断请求标志 **(IF0L, IF0H, IF1L** 和 **IF1H)**。如果产生了中断请求，则停止自编程操作。
**4.** 即使在 **DI** 状态下也可以通过未屏蔽的中断请求停止自编程。要避免这种情况，可以通过使用中断屏蔽标志寄存器**(MK0L, MK0H, MK1L** 和 **MK1H)**屏蔽该中断。
**5.** 将用于自编程的入口程序分配在公共区域**(0000H ～ 7FFFH)**。

**图 25-17. 用于自编程的操作模式与存储器映射(μPD78F0515)**



下图所示使用自编程示例库重写 flash 存储器的流程图。

<R>　**图 25-18.　自编程流程图 (重写 Flash 存储器)**




备注　　对于自编程示例库的详细说明，请参看 **78K0/Kx2 Flash 存储器自编程用户手册 (U17516E)** 。

下表为对于自编程示例库的处理时间和中断响应时间。

<R>

**表 25-13. 对于自编程示例库的处理时间和中断响应时间 (1/4)**

**(1)** 当使用内部高速振荡时钟并且 **RAM** 入口地址在短直接寻址范围之外

| 库名称 | 处理时间 (μs) | | | | 中断响应时间 (μs) | |
|---|---|---|---|---|---|---|
| | C 编译器的普通模式 | | C 编译器/汇编器的静态模式 | | | |
| | 最小 | 最大 | 最小 | 最大 | 最小 | 最大 |
| 自编程开始库 | 4.25 | | | | – | – |
| 初始化库 | 977.75 | | | | – | – |
| 模式校验库 | 753.875 | | 753.125 | | – | – |
| Block 空白校验库 | 12770.875 | | 12765.875 | | 391.25 | 1300.5 |
| Block 擦除库 | 36909.5 | 356318 | 36904.5 | 356296.25 | 389.25 | 1393.5 |
| 字写入库 | 1214<br>(1214.375) | 2409<br>(2409.375) | 1207<br>(1207.375) | 2402<br>(2402.375) | 394.75 | 1289.5 |
| 程序校验库 | 25618.875 | | 25613.875 | | 390.25 | 1324.5 |
| 自编程结束库 | 4.25 | | | | – | – |
| 获取信息库<br>(选项值: 03H) | 871.25<br>(871.375) | | 866<br>(866.125) | | – | – |
| 获取信息库<br>(选项值: 04H) | 863.375<br>(863.5) | | 858.125<br>(858.25) | | – | – |
| 获取信息库<br>(选项值: 05H) | 1024.75<br>(1043.625) | | 1037.5<br>(1038.375) | | – | – |
| 设定信息库 | 105524.75 | 790809.375 | 105523.75 | 790808.375 | 387 | 852.5 |
| EEPROM 写入库 | 1496.5<br>(1496.875) | 2691.5<br>(2691.875) | 1489.5<br>(1489.875) | 2684.5<br>(2684.875) | 399.75 | 1395.5 |

**备注**　括号中的值表示当写入的开始地址结构不在内部高速 RAM 中。

<R> **表 25-13. 对于自编程示例库的处理时间和中断响应时间 (2/4)**

**(2) 当使用内部高速振荡时钟并且 RAM 入口地址在短直接寻址范围 (FE20H) 中**

| 库名称 | 处理时间 (μs) | | | | 中断响应时间 (μs) | |
|---|---|---|---|---|---|---|
| | C 编译器的普通模式 | | C 编译器/汇编器的静态模式 | | | |
| | 最小 | 最大 | 最小 | 最大 | 最小 | 最大 |
| 自编程开始库 | 4.25 | | | | – | – |
| 初始化库 | 443.5 | | | | – | – |
| 模式校验库 | 219.625 | | 218.875 | | – | – |
| Block 空白校验库 | 12236.625 | | 12231.625 | | 81.25 | 727.5 |
| Block 擦除库 | 36363.25 | 355771.75 | 36358.25 | 355750 | 79.25 | 820.5 |
| 字写入库 | 679.75<br>(680.125) | 1874.75<br>(1875.125) | 672.75<br>(673.125) | 1867.75<br>(1868.125) | 84.75 | 716.5 |
| 程序校验库 | 25072.625 | | 25067.625 | | 80.25 | 751.5 |
| 自编程结束库 | 4.25 | | | | – | – |
| 获取信息库<br>(选项值: 03H) | 337<br>(337.125) | | 331.75<br>(331.875) | | – | – |
| 获取信息库<br>(选项值: 04H) | 329.125<br>(239.25) | | 323.875<br>(324) | | – | – |
| 获取信息库<br>(选项值: 05H) | 502.25<br>(503.125) | | 497<br>(497.875) | | – | – |
| 设定信息库 | 104978.5 | 541143.125 | 104977.5 | 541142.125 | 77 | 279.5 |
| EEPROM 写入库 | 962.25<br>(962.625) | 2157.25<br>(2157.625) | 955.25<br>(955.625) | 2150.25<br>(2150.625) | 89.75 | 822.5 |

**备注** 括号中的值表示当写入的开始地址结构不在内部高速 RAM 中。

<R>

**表 25-13. 对于自编程示例库的处理时间和中断响应时间 (3/4)**

**(3)** 当使用高速系统时钟 **(X1** 振荡器或外部时钟输入**)** 并且 **RAM** 入口地址在短直接寻址范围之外

| 库名称 | 处理时间 ($\mu$s) | | | | 中断响应时间 ($\mu$s) | |
|---|---|---|---|---|---|---|
| | C 编译器的普通模式 | | C 编译器/汇编器的静态模式 | | | |
| | 最小 | 最大 | 最小 | 最大 | 最小 | 最大 |
| 自编程开始库 | $34/f_{XH}$ | | | | – | – |
| 初始化库 | $49/f_{XH}$ + 485.8125 | | | | – | – |
| 模式校验库 | $35/f_{XH}$ + 374.75 | | $29/f_{XH}$ + 374.75 | | – | – |
| Block 空白校验库 | $174/f_{XH}$ + 6382.0625 | | $134/f_{XH}$ + 6382.0625 | | $18/f_{XH}$ + 192 | $28/f_{XH}$ + 698 |
| Block 擦除库 | $174/f_{XH}$ + 31093.875 | $174/f_{XH}$ + 298948.125 | $134/f_{XH}$ + 31093.875 | $134/f_{XH}$ + 298948.125 | $18/f_{XH}$ + 186 | $28/f_{XH}$ + 745 |
| 字写入库 | 318 (321)/$f_{XH}$ + 644.125 | 318 (321)/$f_{XH}$ + 1491.625 | 262 (265)/$f_{XH}$ + 644.125 | 262 (265)/$f_{XH}$ + 1491.625 | $22/f_{XH}$ + 189 | $28/f_{XH}$ + 693 |
| 程序校验库 | $174/f_{XH}$ + 13448.5625 | | $134/f_{XH}$ + 13448.5625 | | $18/f_{XH}$ + 192 | $28/f_{XH}$ + 709 |
| 自编程结束库 | $34/f_{XH}$ | | | | – | – |
| 获取信息库 (选项值: 03H) | 171 (172)/$f_{XH}$ + 432.4375 | | 129 (130)/$f_{XH}$ + 432.4375 | | – | – |
| 获取信息库 (选项值: 04H) | 181 (182)/$f_{XH}$ + 427.875 | | 139 (140)/$f_{XH}$ + 427.875 | | – | – |
| 获取信息库 (选项值: 05H) | 404 (411)/$f_{XH}$ + 496.125 | | 362 (369)/$f_{XH}$ + 496.125 | | – | – |
| 设定信息库 | $75/f_{XH}$ + 79157.6875 | $75/f_{XH}$ + 652400 | $67/f_{XH}$ + 79157.6875 | $67/f_{XH}$ + 652400 | $16/f_{XH}$ + 190 | $28/f_{XH}$ + 454 |
| EEPROM 写入库 | 318 (321)/$f_{XH}$ + 799.875 | 318 (321)/$f_{XH}$ + 1647.375 | 262 (265)/$f_{XH}$ + 799.875 | 262 (265)/$f_{XH}$ + 1647.375 | $22/f_{XH}$ + 191 | $28/f_{XH}$ + 783 |

**备注**　**1.** 括号中的值表示当写入的开始地址结构不在内部高速 RAM 中。

**2.** $f_{XH}$: 高速系统时钟频率

<R> 表 **25-13.** 对于自编程示例库的处理时间和中断响应时间 **(4/4)**

**(4)** 当使用高速系统时钟 **(X1** 振荡器或外部时钟输入**)** 并且 **RAM** 入口地址在短直接寻址范围 **(FE20H)** 中

| 库名称 | 处理时间 ($\mu$s) | | | | 中断响应时间 ($\mu$s) | |
|---|---|---|---|---|---|---|
| | C 编译器的普通模式 | | C 编译器/汇编器的静态模式 | | | |
| | 最小 | 最大 | 最小 | 最大 | 最小 | 最大 |
| 自编程开始库 | 34/$f_{XH}$ | | | | – | – |
| 初始化库 | 49/$f_{XH}$ + 224.6875 | | | | – | – |
| 模式校验库 | 35/$f_{XH}$ + 113.625 | | 29/$f_{XH}$ + 113.625 | | – | – |
| Block 空白校验库 | 174/$f_{XH}$ + 6120.9375 | | 134/$f_{XH}$ + 6120.9375 | | 18/$f_{XH}$ + 55 | 28/$f_{XH}$ + 462 |
| Block 擦除库 | 174/$f_{XH}$ + 30820.75 | 174/$f_{XH}$ + 298675 | 134/$f_{XH}$ + 30820.75 | 134/$f_{XH}$ + 298675 | 18/$f_{XH}$ + 49 | 28/$f_{XH}$ + 509 |
| 字写入库 | 318 (321)/$f_{XH}$ + 383 | 318 (321)/$f_{XH}$ + 1230.5 | 262 (265)/$f_{XH}$ + 383 | 262 (265)/$f_{XH}$ + 1230.5 | 22/$f_{XH}$ + 52 | 28/$f_{XH}$ + 457 |
| 程序校验库 | 174/$f_{XH}$ + 13175.4375 | | 134/$f_{XH}$ + 13175.4375 | | 18/$f_{XH}$ + 55 | 28/$f_{XH}$ + 473 |
| 自编程结束库 | 34/$f_{XH}$ | | | | – | – |
| 获取信息库 (选项值: 03H) | 171 (172)/$f_{XH}$ + 171.3125 | | 129 (130)/$f_{XH}$ + 171.3125 | | – | – |
| 获取信息库 (选项值: 04H) | 181 (182)/$f_{XH}$ + 166.75 | | 139 (140)/$f_{XH}$ + 166.75 | | – | – |
| 获取信息库 (选项值: 05H) | 404 (411)/$f_{XH}$ + 231.875 | | 362 (369)/$f_{XH}$ + 231.875 | | – | – |
| 设定信息库 | 75/$f_{XH}$ + 78884.5625 | 75/$f_{XH}$ + 527566.875 | 67/$f_{XH}$ + 78884.5625 | 67/$f_{XH}$ + 527566.875 | 16/$f_{XH}$ +53 | 28/$f_{XH}$ +218 |
| EEPROM 写入库 | 318 (321)/$f_{XH}$ + 538.75 | 318 (321)/fxH + 1386.25 | 262 (265)/$f_{XH}$ + 538.75 | 262 (265)/$f_{XH}$ + 1386.25 | 22/$f_{XH}$ +54 | 28/$f_{XH}$ +547 |

备注 **1.** 括号中的值表示当写入的开始地址结构不在内部高速 RAM 中。

**2.** $f_{XH}$: 高速系统时钟频率

### 25.10.1 引导交换功能

如果在自编程期间由于电源失效或其它原因导致重写引导区域的操作失败，则引导区域的数据可能会丢失且不能通过复位重启程序。

使用引导交换功能可以避免这种情况。

在擦除引导簇 0(这是一个引导程序区域)[注]之前，先通过自编程将一个新的引导程序写入引导簇 1。当将该程序正确写入引导簇 1 后，通过使用 78K0/KC2 固件的设置信息功能将引导簇 1 与引导簇 0 的内容交换，这样引导簇 1 就用作引导区域。之后对初始引导程序区域，即引导簇 0 进行擦除或写操作。

这样，即使在重写引导程序区域时出现电源失效，也能够正确执行程序，这是因为程序在复位并再次执行时实行了从引导簇 1 的引导交换。

如果已经将程序正确写入引导簇 0，则通过使用 78K0/KC2 固件的设置信息功能恢复初始引导区域。

**注** 一个引导簇的容量为 4 KB，通过引导交换功能交换引导簇 0 与引导簇 1 的内容。

引导簇 0 (0000H ～ 0FFFH): 初始引导程序区域

引导簇 1 (1000H ～ 1FFFH): 用于引导交换的区域

<R> **图 25-19. 启动交换功能**

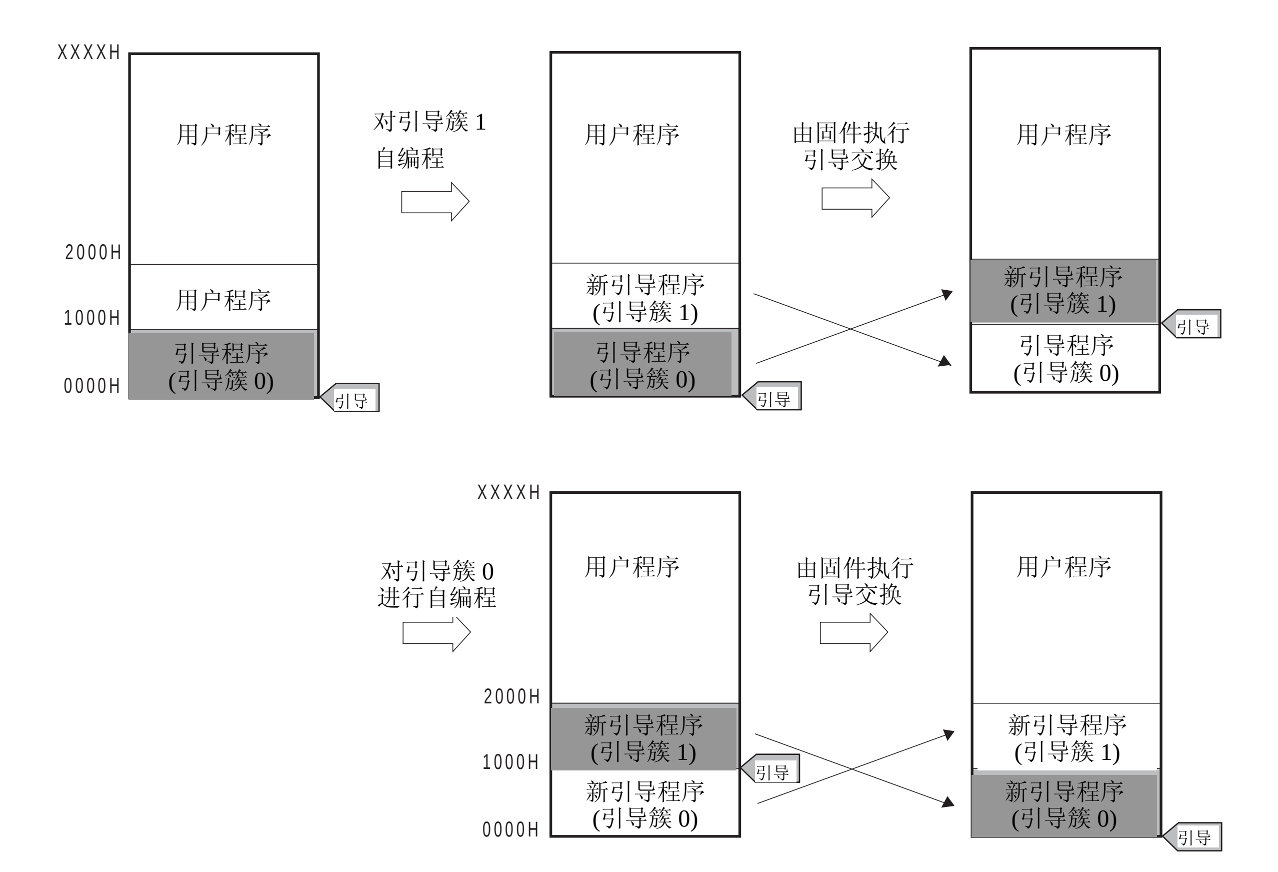
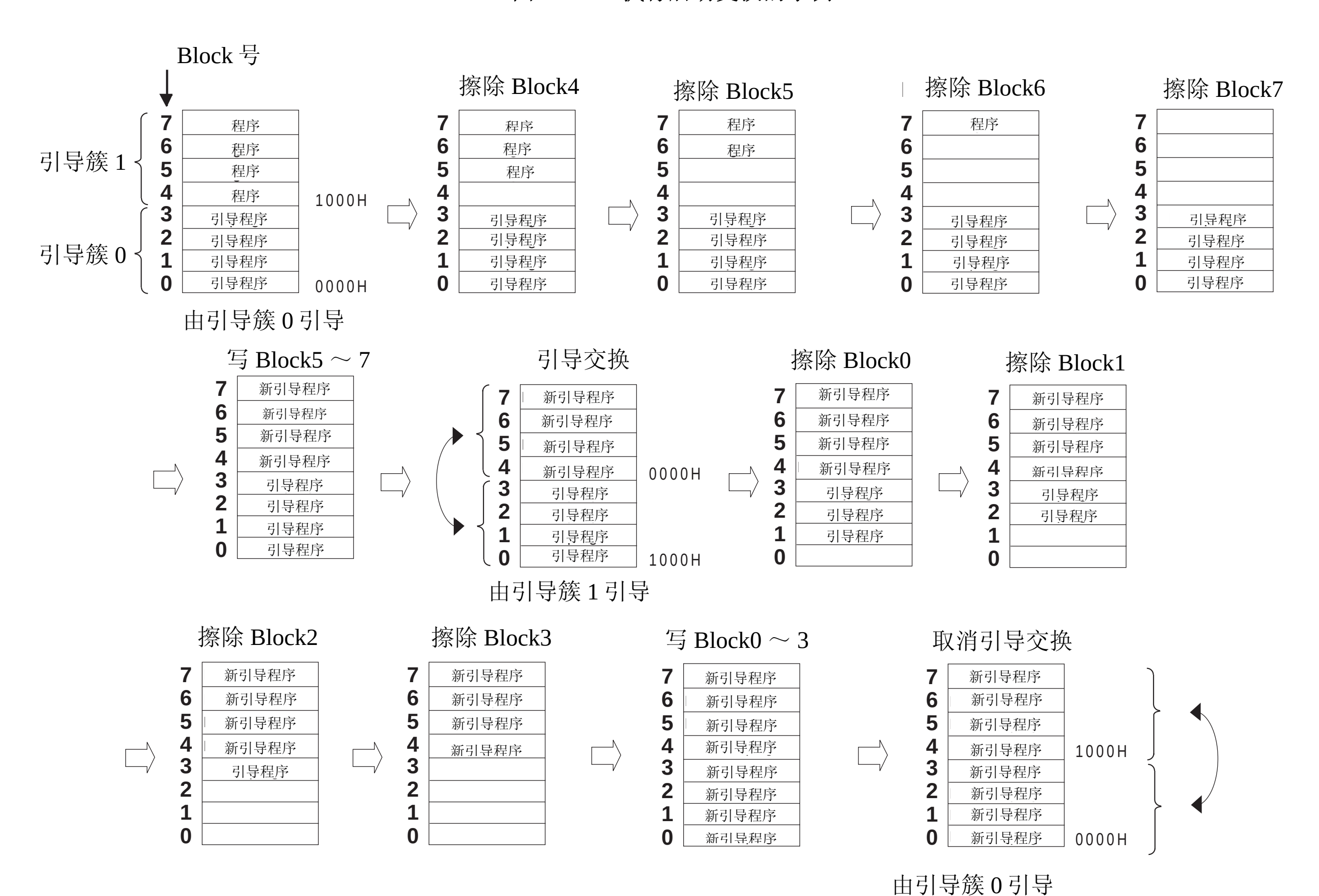


图 25-20. 执行启动交换的示例



## 26.1 μPD78F0513D 和 78F0515D 与 QB-78K0MINI 的连接

μPD78F0513D 和 78F0515D 使用 $V_{DD}$、FLMD0、$\overline{\text{RESET}}$、OCD0A/X1（或 OCD1A/P31）、OCD0B/X2（或 OCD1B/P32),和 $V_{SS}$ 引脚，通过片上调试仿真器（QB-78K0MINI）与主机通信，进行片上调试。可以选择 OCD0A/X1 和 OCD1A/P31 或 OCD0B/X2 和 OCD1B/P32。

注意事项 **μPD78F0513D 和 78F0515D** 具有片上调试功能。由于在使用片上调试功能后产品的稳定性不能得到保证，因此在给定 **flash** 存储器可重写次数的情况下不要将该产品用于大规模生产。**NEC Electronics** 不接受有关该产品的投诉。

**图 26-1. QB-78K0MINI 与μPD78F0513D, 78F0515D 的连接示例**
**(当使用 OCD0A/X1 和 OCD0B/X2 时)**

QB-78K0MINI 目标连接器
μPD78F0513D, 78F0515D
FLMD0
FLMD0
注 1
RESET_IN
目标复位
$\overline{\text{RESET_OUT}}$
$\overline{\text{RESET}}$
X1
OCD0A/X1
X2
OCD0B/X2
GND
GND
$V_{DD}$
$V_{DD}$
P31
注 1

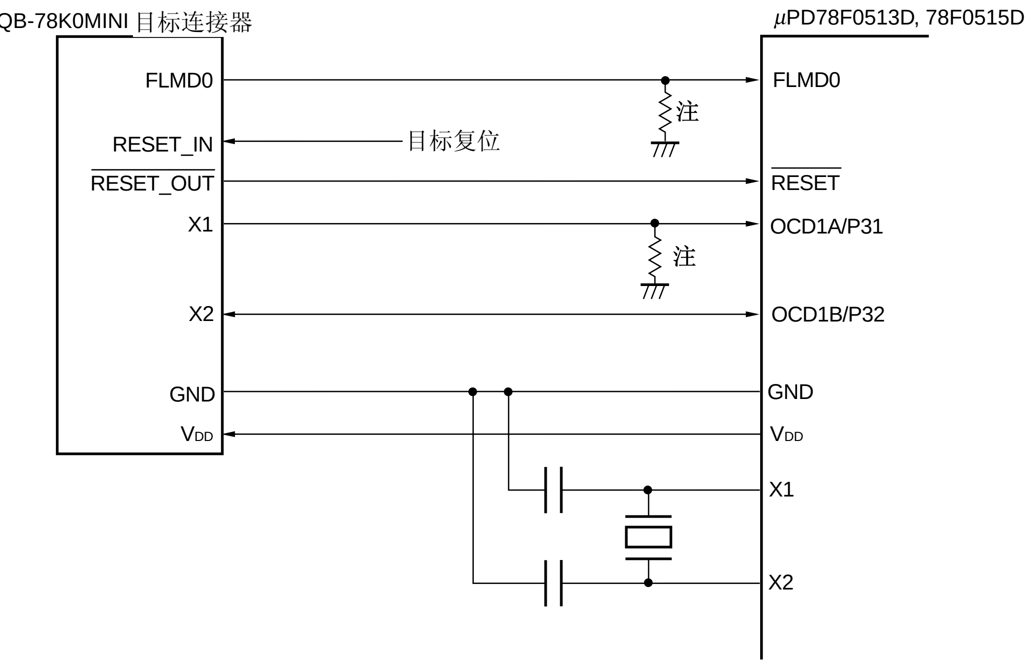
注意事项 **1.** 在片上调试期间从 **OCD0A/X1** 引脚输入时钟。

**2.** 通过外部下拉**OCD1A/P31** 引脚或通过使用**P130** 注 2引脚的外部电路(当该设备复位时输出低电平)，控制**OCD0A/X1** 和**OCD0B/X2** 引脚 。

<R> 注 **1.** 下拉电阻值至少应为 470 Ω。(推荐 10 kΩ)

**2.** 仅 48 引脚产品。

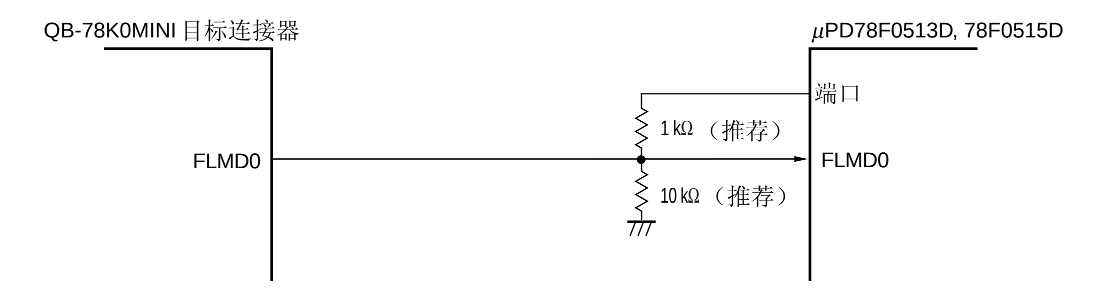
**图 26-2. QB-78K0MINI 与μPD78F0513D, 78F0515D 连接示例**
**(当使用 OCD1A/P31 和 OCD1B/P32 时)**




<R>　注　下拉电阻值至少应为 470 Ω(推荐 10 kΩ)。

当通过片上调试功能执行自编程时按以下连接 FLMD0 引脚。

<R> **图 26-3. 通过片上调试功能执行自编程时连接 FLMD0 引脚**



## 26.2 片上调试安全 ID

在μPD78F0513D 和 78F0515D 中，一个片上调试操作控制标志存放在 flash 存储器地址 0084H 中 (参见**第二十四章 选项字节**)，片上调试安全 ID 设置在地址 0085H ～ 008EH 中。

当使用引导交换功能时，由于 0084H、0085H ～ 008EH 与 1084H 以及 1085H ～ 108EH 的内容被交换，因此预先设置与 1084H 以及 1085H ～ 108EH 的内容相同的值。

如需了解片上调试安全 ID 的详细内容，可参见 QB-78K0MINI 用户手册 (U17029E)。

**表 26-1. 片上调试安全 ID**

| 地址 | 片上调试安全 ID |
|---|---|
| 0085H ～ 008EH | 10 字节的 ID 代码 |
| 1085H ～ 108EH | |

# 第二十七章　指令集

本章以表格方式列出了 78K0/KC2 的指令集。如需了解每种指令的操作和操作代码，可参阅 **78K/0 系列指令用户手册（U12326E）**。

## 27.1 操作列表使用规则

### 27.1.1 操作数标识符和标识方法

根据规范确定的指令操作数标识方法（详情可参见汇编程序编程规范），在每种指令的“操作数”栏列出操作数。如果有两种或两种以上的标识方法，可选其中之一。大写字母和符号#、!、$ 和[ ]是关键字，必须按其原样书写。每种符号的含义如下所示。

- #:　立即数标识
- !:　绝对地址标识
- $:　相对地址标识
- [ ]:间接地址标识

立即数用来描述一个数值型数据或标号。当使用标号时，注意必须加上符号#, !, $, 和[ ]。

对应操作数寄存器标识符 r 和 rp，功能名称 （X, A, C, 等）或绝对名称 （下表括号中的名称： R0, R1, R2 等）都可用于标识。

**表 27-1. 操作数标识符和标识方法**

| 标识符 | 标识方法 |
|---|---|
| r<br>rp<br>sfr<br>sfrp | X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)<br>AX (RP0), BC (RP1), DE (RP2), HL (RP3)<br>特殊功能寄存器符号[注]<br>特殊功能寄存器符号 （仅用于 16 位可操作寄存器偶地址）[注] |
| saddr<br>saddrp | FE20H ～FF1FH 立即数或标号<br>FE20H ～ FF1FH 立即数或标号 （仅用于偶地址） |
| addr16<br><br>addr11<br>addr5 | 0000H ～ FFFFH 立即数或标号<br>（仅用于 16 位数据传送指令的偶地址）<br>0800H ～ 0FFFH 立即数或标号<br>0040H ～ 007FH 立即数或标号 （仅用于偶地址） |
| word<br>byte<br>bit | 16 位立即数或标号<br>8 位立即数或标号<br>3 位立即数或标号 |
| RBn | RB0 ～ RB3 |

**注**　不能使用这些操作数访问地址 FFD0H ～ FFDFH。

**备注**　特殊功能寄存器符号参见**表 3-7 特殊功能寄存器列表** 。

### 27.1.2 操作栏描述

A: A 寄存器; 8 位累加器
X: X 寄存器
B: B 寄存器
C: C 寄存器
D: D 寄存器
E: E 寄存器
H: H 寄存器
L: L 寄存器
AX: AX 寄存器对; 16 位累加器
BC: BC 寄存器对
DE: DE 寄存器对
HL: HL 寄存器对
PC: 程序计数器
SP: 堆栈指针
PSW: 程序状态字
CY: 进位标志
AC: 辅助进位标志
Z: 零标志
RBS: 寄存器 bank 选择标志
IE: 中断请求允许标志
( ): 括号中的地址或寄存器所指的存储单元的内容
$X_H$, $X_L$: 16 位寄存器的高 8 位和低 8 位
∧: 逻辑与 (AND)
∨: 逻辑或 (OR)
$\veebar$: 逻辑异或 (exclusive OR)
$\overline{\quad}$: 数据取反
addr16: 16 位立即数或标号
jdisp8: 带符号的 8 位数据（偏移量）

### 27.1.3 标志操作栏的描述

(空): 不受影响
0: 清零
1: 设置为 1
×: 根据结果进行设置/清零
R: 恢复先前保存的值

## 27.2 操作列表

| 指令组 | 助记符 | 操作数 | 字节 | 时钟 | | 操作 | 标志 |
|---|---|---|---|---|---|---|---|
| | | | | 注 1 | 注 2 | | Z AC CY |
| 8 位数据传送 | **MOV** | r, #byte | 2 | 4 | – | r ← byte | |
| | | saddr, #byte | 3 | 6 | 7 | (saddr) ← byte | |
| | | sfr, #byte | 3 | – | 7 | sfr ← byte | |
| | | A, r 注 3 | 1 | 2 | – | A ← r | |
| | | r, A 注 3 | 1 | 2 | – | r ← A | |
| | | A, saddr | 2 | 4 | 5 | A ← (saddr) | |
| | | saddr, A | 2 | 4 | 5 | (saddr) ← A | |
| | | A, sfr | 2 | – | 5 | A ← sfr | |
| | | sfr, A | 2 | – | 5 | sfr ← A | |
| | | A, !addr16 | 3 | 8 | 9 | A ← (addr16) | |
| | | !addr16, A | 3 | 8 | 9 | (addr16) ← A | |
| | | PSW, #byte | 3 | – | 7 | PSW ← byte | × × × |
| | | A, PSW | 2 | – | 5 | A ← PSW | |
| | | PSW, A | 2 | – | 5 | PSW ← A | × × × |
| | | A, [DE] | 1 | 4 | 5 | A ← (DE) | |
| | | [DE], A | 1 | 4 | 5 | (DE) ← A | |
| | | A, [HL] | 1 | 4 | 5 | A ← (HL) | |
| | | [HL], A | 1 | 4 | 5 | (HL) ← A | |
| | | A, [HL + byte] | 2 | 8 | 9 | A ← (HL + byte) | |
| | | [HL + byte], A | 2 | 8 | 9 | (HL + byte) ← A | |
| | | A, [HL + B] | 1 | 6 | 7 | A ← (HL + B) | |
| | | [HL + B], A | 1 | 6 | 7 | (HL + B) ← A | |
| | | A, [HL + C] | 1 | 6 | 7 | A ← (HL + C) | |
| | | [HL + C], A | 1 | 6 | 7 | (HL + C) ← A | |
| | **XCH** | A, r 注 3 | 1 | 2 | – | A ↔ r | |
| | | A, saddr | 2 | 4 | 6 | A ↔ (saddr) | |
| | | A, sfr | 2 | – | 6 | A ↔ (sfr) | |
| | | A, !addr16 | 3 | 8 | 10 | A ↔ (addr16) | |
| | | A, [DE] | 1 | 4 | 6 | A ↔ (DE) | |
| | | A, [HL] | 1 | 4 | 6 | A ↔ (HL) | |
| | | A, [HL + byte] | 2 | 8 | 10 | A ↔ (HL + byte) | |
| | | A, [HL + B] | 2 | 8 | 10 | A ↔ (HL + B) | |
| | | A, [HL + C] | 2 | 8 | 10 | A ↔ (HL + C) | |

注 **1.** 当访问内部高速 RAM 时或针对无数据访问的指令
**2.** 当访问内部高速 RAM 以外的区域时
**3.** “r = A”除外

备注 **1.** 一个指令时钟周期是指由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（$f_{CPU}$）的一个周期。
**2.** 该时钟周期用于内部 ROM 程序。

| 指令组 | 助记符 | 操作数 | 字节 | 时钟 | | 操作 | 标志 |
|---|---|---|---|---|---|---|---|
| | | | | 注 1 | 注 2 | | Z AC CY |
| 16 位数据传送 | **MOVW** | rp, #word | 3 | 6 | – | rp ← word | |
| | | saddrp, #word | 4 | 8 | 10 | (saddrp) ← word | |
| | | sfrp, #word | 4 | – | 10 | sfrp ← word | |
| | | AX, saddrp | 2 | 6 | 8 | AX ← (saddrp) | |
| | | saddrp, AX | 2 | 6 | 8 | (saddrp) ← AX | |
| | | AX, sfrp | 2 | – | 8 | AX ← sfrp | |
| | | sfrp, AX | 2 | – | 8 | sfrp ← AX | |
| | | AX, rp 注 3 | 1 | 4 | – | AX ← rp | |
| | | rp, AX 注 3 | 1 | 4 | – | rp ← AX | |
| | | AX, !addr16 | 3 | 10 | 12 | AX ← (addr16) | |
| | | !addr16, AX | 3 | 10 | 12 | (addr16) ← AX | |
| | **XCHW** | AX, rp 注 3 | 1 | 4 | – | AX ↔ rp | |
| 8 位操作 | **ADD** | A, #byte | 2 | 4 | – | A, CY ← A + byte | × × × |
| | | saddr, #byte | 3 | 6 | 8 | (saddr), CY ← (saddr) + byte | × × × |
| | | A, r 注 4 | 2 | 4 | – | A, CY ← A + r | × × × |
| | | r, A | 2 | 4 | – | r, CY ← r + A | × × × |
| | | A, saddr | 2 | 4 | 5 | A, CY ← A + (saddr) | × × × |
| | | A, !addr16 | 3 | 8 | 9 | A, CY ← A + (addr16) | × × × |
| | | A, [HL] | 1 | 4 | 5 | A, CY ← A + (HL) | × × × |
| | | A, [HL + byte] | 2 | 8 | 9 | A, CY ← A + (HL + byte) | × × × |
| | | A, [HL + B] | 2 | 8 | 9 | A, CY ← A + (HL + B) | × × × |
| | | A, [HL + C] | 2 | 8 | 9 | A, CY ← A + (HL + C) | × × × |
| | **ADDC** | A, #byte | 2 | 4 | – | A, CY ← A + byte + CY | × × × |
| | | saddr, #byte | 3 | 6 | 8 | (saddr), CY ← (saddr) + byte + CY | × × × |
| | | A, r 注 4 | 2 | 4 | – | A, CY ← A + r + CY | × × × |
| | | r, A | 2 | 4 | – | r, CY ← r + A + CY | × × × |
| | | A, saddr | 2 | 4 | 5 | A, CY ← A + (saddr) + CY | × × × |
| | | A, !addr16 | 3 | 8 | 9 | A, CY ← A + (addr16) + C | × × × |
| | | A, [HL] | 1 | 4 | 5 | A, CY ← A + (HL) + CY | × × × |
| | | A, [HL + byte] | 2 | 8 | 9 | A, CY ← A + (HL + byte) + CY | × × × |
| | | A, [HL + B] | 2 | 8 | 9 | A, CY ← A + (HL + B) + CY | × × × |
| | | A, [HL + C] | 2 | 8 | 9 | A, CY ← A + (HL + C) + CY | × × × |

注 **1.** 当访问内部高速 RAM 时或针对无数据访问的指令
**2.** 当访问内部高速 RAM 以外的区域时
**3.** 仅当 rp = BC, DE 或 HL
**4.** “r = A”除外

备注 **1.** 一个指令时钟周期是指由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（$f_{CPU}$）的一个周期。
**2.** 该时钟周期用于内部 ROM 程序。

| 指令组 | 助记符 | 操作数 | 字节 | 时钟 | | 操作 | 标志 |
|---|---|---|---|---|---|---|---|
| | | | | 注 1 | 注 2 | | Z AC CY |
| 8 位操作 | **SUB** | A, #byte | 2 | 4 | – | A, CY ← A – byte | × × × |
| | | saddr, #byte | 3 | 6 | 8 | (saddr), CY ← (saddr) – byte | × × × |
| | | A, r 注 3 | 2 | 4 | – | A, CY ← A – r | × × × |
| | | r, A | 2 | 4 | – | r, CY ← r – A | × × × |
| | | A, saddr | 2 | 4 | 5 | A, CY ← A – (saddr) | × × × |
| | | A, !addr16 | 3 | 8 | 9 | A, CY ← A – (addr16) | × × × |
| | | A, [HL] | 1 | 4 | 5 | A, CY ← A – (HL) | × × × |
| | | A, [HL + byte] | 2 | 8 | 9 | A, CY ← A – (HL + byte) | × × × |
| | | A, [HL + B] | 2 | 8 | 9 | A, CY ← A – (HL + B) | × × × |
| | | A, [HL + C] | 2 | 8 | 9 | A, CY ← A – (HL + C) | × × × |
| | **SUBC** | A, #byte | 2 | 4 | – | A, CY ← A – byte – CY | × × × |
| | | saddr, #byte | 3 | 6 | 8 | (saddr), CY ← (saddr) – byte – CY | × × × |
| | | A, r 注 3 | 2 | 4 | – | A, CY ← A – r – CY | × × × |
| | | r, A | 2 | 4 | – | r, CY ← r – A – CY | × × × |
| | | A, saddr | 2 | 4 | 5 | A, CY ← A – (saddr) – CY | × × × |
| | | A, !addr16 | 3 | 8 | 9 | A, CY ← A – (addr16) – CY | × × × |
| | | A, [HL] | 1 | 4 | 5 | A, CY ← A – (HL) – CY | × × × |
| | | A, [HL + byte] | 2 | 8 | 9 | A, CY ← A – (HL + byte) – CY | × × × |
| | | A, [HL + B] | 2 | 8 | 9 | A, CY ← A – (HL + B) – CY | × × × |
| | | A, [HL + C] | 2 | 8 | 9 | A, CY ← A – (HL + C) – CY | × × × |
| | **AND** | A, #byte | 2 | 4 | – | A ← A ∧ byte | × |
| | | saddr, #byte | 3 | 6 | 8 | (saddr) ← (saddr) ∧ byte | × |
| | | A, r 注 3 | 2 | 4 | – | A ← A ∧ r | × |
| | | r, A | 2 | 4 | – | r ← r ∧ A | × |
| | | A, saddr | 2 | 4 | 5 | A ← A ∧ (saddr) | × |
| | | A, !addr16 | 3 | 8 | 9 | A ← A ∧ (addr16) | × |
| | | A, [HL] | 1 | 4 | 5 | A ← A ∧ (HL) | × |
| | | A, [HL + byte] | 2 | 8 | 9 | A ← A ∧ (HL + byte) | × |
| | | A, [HL + B] | 2 | 8 | 9 | A ← A ∧ (HL + B) | × |
| | | A, [HL + C] | 2 | 8 | 9 | A ← A ∧ (HL + C) | × |

注　**1.** 当访问内部高速 RAM 或针对无数据访问的指令
　　**2.** 当访问内部高速 RAM 以外的区域时
　　**3.** “r = A”除外

备注　**1.** 一个指令时钟周期是指由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（$f_{CPU}$）的一个周期。
　　　**2.** 该时钟周期用于内部 ROM 程序。

| 指令组 | 助记符 | 操作数 | 字节数 | 时钟 | | 操作 | 标志 | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | 注 1 | 注 2 | | Z | AC | CY |
| 8 位操作 | **OR** | A, #byte | 2 | 4 | – | A ← A ∨ byte | × | | |
| | | saddr, #byte | 3 | 6 | 8 | (saddr) ← (saddr) ∨ byte | × | | |
| | | A, r [注 3] | 2 | 4 | – | A ← A ∨ r | × | | |
| | | r, A | 2 | 4 | – | r ← r ∨ A | × | | |
| | | A, saddr | 2 | 4 | 5 | A ← A ∨ (saddr) | × | | |
| | | A, !addr16 | 3 | 8 | 9 | A ← A ∨ (addr16) | × | | |
| | | A, [HL] | 1 | 4 | 5 | A ← A ∨ (HL) | × | | |
| | | A, [HL + byte] | 2 | 8 | 9 | A ← A ∨ (HL + byte) | × | | |
| | | A, [HL + B] | 2 | 8 | 9 | A ← A ∨ (HL + B) | × | | |
| | | A, [HL + C] | 2 | 8 | 9 | A ← A ∨ (HL + C) | × | | |
| | **XOR** | A, #byte | 2 | 4 | – | A ← A ⊻ byte | × | | |
| | | saddr, #byte | 3 | 6 | 8 | (saddr) ← (saddr) ⊻ byte | × | | |
| | | A, r [注 3] | 2 | 4 | – | A ← A ⊻ r | × | | |
| | | r, A | 2 | 4 | – | r ← r ⊻ A | × | | |
| | | A, saddr | 2 | 4 | 5 | A ← A ⊻ (saddr) | × | | |
| | | A, !addr16 | 3 | 8 | 9 | A ← A ⊻ (addr16) | × | | |
| | | A, [HL] | 1 | 4 | 5 | A ← A ⊻ (HL) | × | | |
| | | A, [HL + byte] | 2 | 8 | 9 | A ← A ⊻ (HL + byte) | × | | |
| | | A, [HL + B] | 2 | 8 | 9 | A ← A ⊻ (HL + B) | × | | |
| | | A, [HL + C] | 2 | 8 | 9 | A ← A ⊻ (HL + C) | × | | |
| | **CMP** | A, #byte | 2 | 4 | – | A – byte | × | × | × |
| | | saddr, #byte | 3 | 6 | 8 | (saddr) – byte | × | × | × |
| | | A, r [注 3] | 2 | 4 | – | A – r | × | × | × |
| | | r, A | 2 | 4 | – | r – A | × | × | × |
| | | A, saddr | 2 | 4 | 5 | A – (saddr) | × | × | × |
| | | A, !addr16 | 3 | 8 | 9 | A – (addr16) | × | × | × |
| | | A, [HL] | 1 | 4 | 5 | A – (HL) | × | × | × |
| | | A, [HL + byte] | 2 | 8 | 9 | A – (HL + byte) | × | × | × |
| | | A, [HL + B] | 2 | 8 | 9 | A – (HL + B) | × | × | × |
| | | A, [HL + C] | 2 | 8 | 9 | A – (HL + C) | × | × | × |

注　**1.** 当访问内部高速 RAM 或针对无数据访问的指令

　　**2.** 当访问内部高速 RAM 以外的区域时

　　**3.** “r = A”除外

备注　**1.** 一个指令时钟周期是指由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（$f_{CPU}$）的一个周期。

　　　**2.** 该时钟周期用于内部 ROM 程序。

| 指令组 | 助记符 | 操作数 | 字节 | 时钟 | | 操作 | 标志 | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | **注 1** | **注 2** | | Z | AC | CY |
| 16 位操作 | **ADDW** | AX, #word | 3 | 6 | – | AX, CY ← AX + word | × | × | × |
| | **SUBW** | AX, #word | 3 | 6 | – | AX, CY ← AX – word | × | × | × |
| | **CMPW** | AX, #word | 3 | 6 | – | AX – word | × | × | × |
| 乘法/除法 | **MULU** | X | 2 | 16 | – | AX ← A × X | | | |
| | **DIVUW** | C | 2 | 25 | – | AX (商), C (余数) ← AX ÷ C | | | |
| 递增/递减 | **INC** | r | 1 | 2 | – | r ← r + 1 | × | × | |
| | | saddr | 2 | 4 | 6 | (saddr) ← (saddr) + 1 | × | × | |
| | **DEC** | r | 1 | 2 | – | r ← r – 1 | × | × | |
| | | saddr | 2 | 4 | 6 | (saddr) ← (saddr) – 1 | × | × | |
| | **INCW** | rp | 1 | 4 | – | rp ← rp + 1 | | | |
| | **DECW** | rp | 1 | 4 | – | rp ← rp – 1 | | | |
| 循环 | **ROR** | A, 1 | 1 | 2 | – | (CY, $A_7$ ← $A_0$, $A_{m-1}$ ← $A_m$) × 1 time | | | × |
| | **ROL** | A, 1 | 1 | 2 | – | (CY, $A_0$ ← $A_7$, $A_{m+1}$ ← $A_m$) × 1 time | | | × |
| | **RORC** | A, 1 | 1 | 2 | – | (CY ← $A_0$, $A_7$ ← CY, $A_{m-1}$ ← $A_m$) × 1 time | | | × |
| | **ROLC** | A, 1 | 1 | 2 | – | (CY ← $A_7$, $A_0$ ← CY, $A_{m+1}$ ← $A_m$) × 1 time | | | × |
| | **ROR4** | [HL] | 2 | 10 | 12 | $A_{3-0}$ ← $(HL)_{3-0}$, $(HL)_{7-4}$ ← $A_{3-0}$, $(HL)_{3-0}$ ← $(HL)_{7-4}$ | | | |
| | **ROL4** | [HL] | 2 | 10 | 12 | $A_{3-0}$ ← $(HL)_{7-4}$, $(HL)_{3-0}$ ← $A_{3-0}$, $(HL)_{7-4}$ ← $(HL)_{3-0}$ | | | |
| BCD 调整 | **ADJBA** | | 2 | 4 | – | 加法后的十进制调整 | × | × | × |
| | **ADJBS** | | 2 | 4 | – | 减法后的十进制调整 | × | × | × |
| 位操作 | **MOV1** | CY, saddr.bit | 3 | 6 | 7 | CY ← (saddr.bit) | | | × |
| | | CY, sfr.bit | 3 | – | 7 | CY ← sfr.bit | | | × |
| | | CY, A.bit | 2 | 4 | – | CY ← A.bit | | | × |
| | | CY, PSW.bit | 3 | – | 7 | CY ← PSW.bit | | | × |
| | | CY, [HL].bit | 2 | 6 | 7 | CY ← (HL).bit | | | × |
| | | saddr.bit, CY | 3 | 6 | 8 | (saddr.bit) ← CY | | | |
| | | sfr.bit, CY | 3 | – | 8 | sfr.bit ← CY | | | |
| | | A.bit, CY | 2 | 4 | – | A.bit ← CY | | | |
| | | PSW.bit, CY | 3 | – | 8 | PSW.bit ← CY | × | × | |
| | | [HL].bit, CY | 2 | 6 | 8 | (HL).bit ← CY | | | |

注　**1.** 当访问内部高速 RAM 或针对无数据访问的指令
　　**2.** 当访问内部高速 RAM 以外的区域时

备注　**1.** 一个指令时钟周期是指由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（$f_{CPU}$）的一个周期。
　　　**2.** 该时钟周期用于内部 ROM 程序。

| 指令组 | 助记符 | 操作数 | 字节 | 时钟 | | 操作 | 标志 | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | **注 1** | **注 2** | | Z | AC | CY |
| 位操作 | **AND1** | CY, saddr.bit | 3 | 6 | 7 | CY ← CY ∧ (saddr.bit) | | | × |
| | | CY, sfr.bit | 3 | – | 7 | CY ← CY ∧ sfr.bit | | | × |
| | | CY, A.bit | 2 | 4 | – | CY ← CY ∧ A.bit | | | × |
| | | CY, PSW.bit | 3 | – | 7 | CY ← CY ∧ PSW.bit | | | × |
| | | CY, [HL].bit | 2 | 6 | 7 | CY ← CY ∧ (HL).bit | | | × |
| | **OR1** | CY, saddr.bit | 3 | 6 | 7 | CY ← CY ∨ (saddr.bit) | | | × |
| | | CY, sfr.bit | 3 | – | 7 | CY ← CY ∨ sfr.bit | | | × |
| | | CY, A.bit | 2 | 4 | – | CY ← CY ∨ A.bit | | | × |
| | | CY, PSW.bit | 3 | – | 7 | CY ← CY ∨ PSW.bit | | | × |
| | | CY, [HL].bit | 2 | 6 | 7 | CY ← CY ∨ (HL).bit | | | × |
| | **XOR1** | CY, saddr.bit | 3 | 6 | 7 | CY ← CY ⊻ (saddr.bit) | | | × |
| | | CY, sfr.bit | 3 | – | 7 | CY ← CY ⊻ sfr.bit | | | × |
| | | CY, A.bit | 2 | 4 | – | CY ← CY ⊻ A.bit | | | × |
| | | CY, PSW. bit | 3 | – | 7 | CY ← CY ⊻ PSW.bit | | | × |
| | | CY, [HL].bit | 2 | 6 | 7 | CY ← CY ⊻ (HL).bit | | | × |
| | **SET1** | saddr.bit | 2 | 4 | 6 | (saddr.bit) ← 1 | | | |
| | | sfr.bit | 3 | – | 8 | sfr.bit ← 1 | | | |
| | | A.bit | 2 | 4 | – | A.bit ← 1 | | | |
| | | PSW.bit | 2 | – | 6 | PSW.bit ← 1 | × | × | × |
| | | [HL].bit | 2 | 6 | 8 | (HL).bit ← 1 | | | |
| | **CLR1** | saddr.bit | 2 | 4 | 6 | (saddr.bit) ← 0 | | | |
| | | sfr.bit | 3 | – | 8 | sfr.bit ← 0 | | | |
| | | A.bit | 2 | 4 | – | A.bit ← 0 | | | |
| | | PSW.bit | 2 | – | 6 | PSW.bit ← 0 | × | × | × |
| | | [HL].bit | 2 | 6 | 8 | (HL).bit ← 0 | | | |
| | **SET1** | CY | 1 | 2 | – | CY ← 1 | | | 1 |
| | **CLR1** | CY | 1 | 2 | – | CY ← 0 | | | 0 |
| | **NOT1** | CY | 1 | 2 | – | CY ← $\overline{CY}$ | | | × |

注　**1.** 当访问内部高速 RAM 或针对无数据访问的指令

　　**2.** 当访问内部高速 RAM 以外的区域时

备注　**1.** 一个指令时钟周期是指由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（$f_{CPU}$）的一个周期。

　　　**2.** 该时钟周期用于内部 ROM 程序。

| 指令组 | 助记符 | 操作数 | 字节数 | 时钟 | | 操作 | 标志 | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | 注 1 | 注 2 | | Z | AC | CY |
| 调用/返回 | **CALL** | !addr16 | 3 | 7 | – | $(SP-1) \leftarrow (PC+3)_H$, $(SP-2) \leftarrow (PC+3)_L$,<br>PC ← addr16, SP ← SP – 2 | | | |
| | **CALLF** | !addr11 | 2 | 5 | – | $(SP-1) \leftarrow (PC+2)_H$, $(SP-2) \leftarrow (PC+2)_L$,<br>$PC_{15-11} \leftarrow 00001$, $PC_{10-0} \leftarrow addr11$,<br>SP ← SP – 2 | | | |
| | **CALLT** | [addr5] | 1 | 6 | – | $(SP-1) \leftarrow (PC+1)_H$, $(SP-2) \leftarrow (PC+1)_L$,<br>$PC_H \leftarrow (00000000, addr5+1)$,<br>$PC_L \leftarrow (00000000, addr5)$,<br>SP ← SP – 2 | | | |
| | **BRK** | | 1 | 6 | – | (SP – 1) ← PSW, $(SP-2) \leftarrow (PC+1)_H$,<br>$(SP-3) \leftarrow (PC+1)_L$, $PC_H \leftarrow (003FH)$,<br>$PC_L \leftarrow (003EH)$, SP ← SP – 3, IE ← 0 | | | |
| | **RET** | | 1 | 6 | – | $PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$,<br>SP ← SP + 2 | | | |
| | **RETI** | | 1 | 6 | – | $PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$,<br>PSW ← (SP + 2), SP ← SP + 3 | R | R | R |
| | **RETB** | | 1 | 6 | – | $PC_H \leftarrow (SP+1)$, $PC_L \leftarrow (SP)$,<br>PSW ← (SP + 2), SP ← SP + 3 | R | R | R |
| 堆栈操作 | **PUSH** | PSW | 1 | 2 | – | (SP – 1) ← PSW, SP ← SP – 1 | | | |
| | | rp | 1 | 4 | – | $(SP-1) \leftarrow rp_H$, $(SP-2) \leftarrow rp_L$,<br>SP ← SP – 2 | | | |
| | **POP** | PSW | 1 | 2 | – | PSW ← (SP), SP ← SP + 1 | R | R | R |
| | | rp | 1 | 4 | – | $rp_H \leftarrow (SP+1)$, $rp_L \leftarrow (SP)$,<br>SP ← SP + 2 | | | |
| | **MOVW** | SP, #word | 4 | – | 10 | SP ← word | | | |
| | | SP, AX | 2 | – | 8 | SP ← AX | | | |
| | | AX, SP | 2 | – | 8 | AX ← SP | | | |
| 无条件转移 | **BR** | !addr16 | 3 | 6 | – | PC ← addr16 | | | |
| | | $addr16 | 2 | 6 | – | PC ← PC + 2 + jdisp8 | | | |
| | | AX | 2 | 8 | – | PCH ← A, $PC_L \leftarrow X$ | | | |
| 条件转移 | **BC** | $addr16 | 2 | 6 | – | 如果 CY = 1，则 PC ← PC + 2 + jdisp8 | | | |
| | **BNC** | $addr16 | 2 | 6 | – | 如果 CY =0，则 PC ← PC + 2 + jdisp8 | | | |
| | **BZ** | $addr16 | 2 | 6 | – | 如果 Z = 1，则 PC ← PC + 2 + jdisp8 | | | |
| | **BNZ** | $addr16 | 2 | 6 | – | 如果 Z = 0，则 PC ← PC + 2 + jdisp8 | | | |

注　**1.** 当访问内部高速 RAM 或针对无数据访问的指令

　　**2.** 当访问内部高速 RAM 以外的区域时

**备注**　**1.** 一个指令时钟周期是指由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（$f_{CPU}$）的一个周期。

　　　**2.** 该时钟周期用于内部 ROM 程序。

| 指令组 | 助记符 | 操作数 | 字节数 | 时钟 | | 操作 | 标志 |
|---|---|---|---|---|---|---|---|
| | | | | 注 1 | 注 2 | | Z AC CY |
| 条件转移 | **BT** | saddr.bit, $addr16 | 3 | 8 | 9 | 如果(saddr.bit) = 1，则 PC ← PC + 3 + jdisp8 | |
| | | sfr.bit, $addr16 | 4 | – | 11 | 如果 sfr.bit = 1，则 PC ← PC + 4 + jdisp8 | |
| | | A.bit, $addr16 | 3 | 8 | – | 如果 A.bit = 1，则 PC ← PC + 3 + jdisp8 | |
| | | PSW.bit, $addr16 | 3 | – | 9 | 如果 PSW.bit = 1，则 PC ← PC + 3 + jdisp8 | |
| | | [HL].bit, $addr16 | 3 | 10 | 11 | 如果(HL).bit = 1，则 PC ← PC + 3 + jdisp8 | |
| | **BF** | saddr.bit, $addr16 | 4 | 10 | 11 | 如果(saddr.bit) = 0，则 PC ← PC + 4 + jdisp8 | |
| | | sfr.bit, $addr16 | 4 | – | 11 | 如果 sfr.bit = 0，则 PC ← PC + 4 + jdisp8 | |
| | | A.bit, $addr16 | 3 | 8 | – | 如果 A.bit = 0，则 PC ← PC + 3 + jdisp8 | |
| | | PSW.bit, $addr16 | 4 | – | 11 | 如果 PSW. bit = 0，则 PC ← PC + 4 + jdisp8 | |
| | | [HL].bit, $addr16 | 3 | 10 | 11 | 如果 (HL).bit = 0，则 PC ← PC + 3 + jdisp8 | |
| | **BTCLR** | saddr.bit, $addr16 | 4 | 10 | 12 | 如果(saddr.bit) = 1，则 PC ← PC + 4 + jdisp8<br>然后复位(saddr.bit) | |
| | | sfr.bit, $addr16 | 4 | – | 12 | 如果 sfr.bit = 1，则 PC ← PC + 4 + jdisp8<br>然后复位 sfr.bit | |
| | | A.bit, $addr16 | 3 | 8 | – | 如果 A.bit = 1，则 PC ← PC + 3 + jdisp8<br>然后复位 A.bit | |
| | | PSW.bit, $addr16 | 4 | – | 12 | 如果 PSW.bit = 1，则 PC ← PC + 4 + jdisp8<br>然后复位 PSW.bit | × × × |
| | | [HL].bit, $addr16 | 3 | 10 | 12 | 如果(HL).bit = 1，则 PC ← PC + 3 + jdisp8<br>然后复位 (HL).bit | |
| | **DBNZ** | B, $addr16 | 2 | 6 | – | B ← B – 1, 然后<br>如果 B ≠ 0，则 PC ← PC + 2 + jdisp8 | |
| | | C, $addr16 | 2 | 6 | – | C ← C –1, 然后<br>如果 C ≠ 0，则 PC ← PC + 2 + jdisp8 | |
| | | saddr, $addr16 | 3 | 8 | 10 | (saddr) ← (saddr) – 1, 然后<br>如果(saddr) ≠ 0，则 PC ← PC + 3 + jdisp8 | |
| CPU 控制 | **SEL** | RBn | 2 | 4 | – | RBS1, 0 ← n | |
| | **NOP** | | 1 | 2 | – | 无操作 | |
| | **EI** | | 2 | – | 6 | IE ← 1 (允许中断) | |
| | **DI** | | 2 | – | 6 | IE ← 0 (禁止中断) | |
| | **HALT** | | 2 | 6 | – | 设置 HALT 模式 | |
| | **STOP** | | 2 | 6 | – | 设置 STOP 模式 | |

**注** **1.** 当访问内部高速 RAM 或针对无数据访问的指令
**2.** 当访问内部高速 RAM 以外的区域时

**备注** **1.** 一个指令时钟周期是指由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（$f_{CPU}$）的一个周期。
**2.** 该时钟周期用于内部 ROM 程序。

## 27.3 按寻址类型列出指令

### (1) 8 位指令

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

| 第 2 操作数<br>第 1 操作数 | #byte | A | r[注] | sfr | saddr | !addr16 | PSW | [DE] | [HL] | [HL + byte]<br>[HL + B]<br>[HL + C] | $addr16 | 1 | 无 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | ADD<br>ADDC<br>SUB<br>SUBC<br>AND<br>OR<br>XOR<br>CMP | | MOV<br>XCH<br>ADD<br>ADDC<br>SUB<br>SUBC<br>AND<br>OR<br>XOR<br>CMP | MOV<br>XCH | MOV<br>XCH<br>ADD<br>ADDC<br>SUB<br>SUBC<br>AND<br>OR<br>XOR<br>CMP | MOV<br>XCH<br>ADD<br>ADDC<br>SUB<br>SUBC<br>AND<br>OR<br>XOR<br>CMP | MOV | MOV<br>XCH | MOV<br>XCH<br>ADD<br>ADDC<br>SUB<br>SUBC<br>AND<br>OR<br>XOR<br>CMP | MOV<br>XCH<br>ADD<br>ADDC<br>SUB<br>SUBC<br>AND<br>OR<br>XOR<br>CMP | | ROR<br>ROL<br>RORC<br>ROLC | |
| r | MOV | MOV<br>ADD<br>ADDC<br>SUB<br>SUBC<br>AND<br>OR<br>XOR<br>CMP | | | | | | | | | | | INC<br>DEC |
| B, C | | | | | | | | | | | DBNZ | | |
| sfr | MOV | MOV | | | | | | | | | | | |
| saddr | MOV<br>ADD<br>ADDC<br>SUB<br>SUBC<br>AND<br>OR<br>XOR<br>CMP | MOV | | | | | | | | | DBNZ | | INC<br>DEC |
| !addr16 | | MOV | | | | | | | | | | | |
| PSW | MOV | MOV | | | | | | | | | | | PUSH<br>POP |
| [DE] | | MOV | | | | | | | | | | | |
| [HL] | | MOV | | | | | | | | | | | ROR4<br>ROL4 |
| [HL + byte]<br>[HL + B]<br>[HL + C] | | MOV | | | | | | | | | | | |
| X | | | | | | | | | | | | | MULU |
| C | | | | | | | | | | | | | DIVUW |

注 “r = A”除外

**(2) 16 位指令**

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

| 第 2 操作数 / 第 1 操作数 | #word | AX | rp[注] | sfrp | saddrp | !addr16 | SP | 无 |
|---|---|---|---|---|---|---|---|---|
| AX | ADDW<br>SUBW<br>CMPW | | MOVW<br>XCHW | MOVW | MOVW | MOVW | MOVW | |
| rp | MOVW | MOVW[注] | | | | | | INCW<br>DECW<br>PUSH<br>POP |
| sfrp | MOVW | MOVW | | | | | | |
| saddrp | MOVW | MOVW | | | | | | |
| !addr16 | | MOVW | | | | | | |
| SP | MOVW | MOVW | | | | | | |

注　　仅当 rp = BC, DE, HL 时使用

**(3) 位操作指令**

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

| 第 2 操作数 / 第 1 操作数 | A.bit | sfr.bit | saddr.bit | PSW.bit | [HL].bit | CY | $addr16 | 无 |
|---|---|---|---|---|---|---|---|---|
| A.bit | | | | | | MOV1 | BT<br>BF<br>BTCLR | SET1<br>CLR1 |
| sfr.bit | | | | | | MOV1 | BT<br>BF<br>BTCLR | SET1<br>CLR1 |
| saddr.bit | | | | | | MOV1 | BT<br>BF<br>BTCLR | SET1<br>CLR1 |
| PSW.bit | | | | | | MOV1 | BT<br>BF<br>BTCLR | SET1<br>CLR1 |
| [HL].bit | | | | | | MOV1 | BT<br>BF<br>BTCLR | SET1<br>CLR1 |
| CY | MOV1<br>AND1<br>OR1<br>XOR1 | MOV1<br>AND1<br>OR1<br>XOR1 | MOV1<br>AND1<br>OR1<br>XOR1 | MOV1<br>AND1<br>OR1<br>XOR1 | MOV1<br>AND1<br>OR1<br>XOR1 | | | SET1<br>CLR1<br>NOT1 |

**(4)** 调用指令 **/** 转移指令

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

| 第 2 操作数 / 第 1 操作数 | AX | !addr16 | !addr11 | [addr5] | $addr16 |
|---|---|---|---|---|---|
| 基本指令 | BR | CALL<br>BR | CALLF | CALLT | BR<br>BC<br>BNC<br>BZ<br>BNZ |
| 复合指令 | | | | | BT<br>BF<br>BTCLR<br>DBNZ |

**(5)** 其它指令

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

目标产品：  **μPD78F0511, 78F0512, 78F0513, 78F0514, 78F0515, 78F0513D, 78F0515D**

注意事项　**μPD78F0513D** 和 **78F0515D** 具有片上调试功能。不要将该产品用于大规模生产，因为在给定 **flash** 存储器重写次数的条件下，使用片上调试功能后，产品的可靠性得不到保证。**NEC Electronics** 不接受有关该产品的任何投诉。

最大额定值 ($T_A$ = 25°C) (1/2)

| 参数 | 符号 | 条件 | | 额定值 | 单位 |
|---|---|---|---|---|---|
| 电源电压 | $V_{DD}$ | | | –0.5 ～ +6.5 | V |
| | $V_{SS}$ | | | –0.5 ～ +0.3 | V |
| | $AV_{REF}$ | | | –0.5 ～ $V_{DD}$ + 0.3[注1] | V |
| | $AV_{SS}$ | | | –0.5 ～ +0.3 | V |
| 输入电压 | $V_{I1}$ | P00, P01, P10 ～ P17, P20 ～ P27, P30 ～ P33, P40, P41, P70 ～ P73, P74[注2], P75[注2], P120 ～ P124, P140[注2], X1, X2, XT1, XT2, $\overline{RESET}$ | | –0.3 ～ $V_{DD}$ + 0.3[注1] | V |
| | $V_{I2}$ | P60 ～ P63 (N-ch 漏极开路) | | –0.3 ～ +6.5 | V |
| 输出电压 | $V_O$ | | | –0.3 ～ $V_{DD}$ + 0.3[注1] | V |
| 模拟输入电压 | $V_{AN}$ | ANI0 ～ ANI7 | | –0.3 ～ $AV_{REF}$ + 0.3[注1]<br>和 –0.3 ～ $V_{DD}$ + 0.3[注1] | V |
| 输出电流，高 | $I_{OH}$ | 每个引脚 | P00, P01, P10 ～ P17, P30 ～ P33, P40, P41, P70 ～ P73, P74[注2], P75[注2], P120, P130[注2], P140[注2] | –10 | mA |
| | | 所有引脚总和 –80 mA | P00, P01, P40, P41, P120, P130[注2], P140[注2] | –25 | mA |
| | | | P10 ～ P17, P30 ～ P33, P70 ～ P73, P74[注2], P75[注2] | –55 | mA |
| | | 每个引脚 | P20 ～ P27 | –0.5 | mA |
| | | 所有引脚总和 | | –2 | mA |
| | | 每个引脚 | P121 ～ P124 | –1 | mA |
| | | 所有引脚总和 | | –4 | mA |

注　**1.**　必须小于等于 6.5 V 。

　　**2.**　仅 48 引脚产品。

**注意事项**　**任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因而，必须保证产品在不超过最大额定值的条件下使用。**

**备注**　除非另外说明，复用功能引脚的特性与端口引脚的相同。

标准产品

**最大额定值 ($T_A$ = 25°C) (2/2)**

| 参数 | 符号 | 条件 | | 额定值 | 单位 |
|---|---|---|---|---|---|
| 输出电流，低 | $I_{OL}$ | 每个引脚 | P00, P01, P10 ～ P17, P30 ～ P33, P40, P41, P60 ～ P63, P70 ～ P73, P74[注], P75[注], P120, P130[注], P140[注] | 30 | mA |
| | | 所有引脚总和 –200 mA | P00, P01, P40, P41, P120, P130[注], P140[注] | 60 | mA |
| | | | P10 ～ P17, P30 ～ P33, P60 ～ P63, P70～ P73, P74[注], P75[注] | 140 | mA |
| <R> | | 每个引脚 | P20 ～ P27 | 1 | mA |
| <R> | | 所有引脚总和 | | 5 | mA |
| <R> | | 每个引脚 | P121 ～ P124 | 4 | mA |
| <R> | | 所有引脚总和 | | 10 | mA |
| 工作环境温度 | $T_A$ | 正常操作模式 | | –40 ～ +85 | °C |
| | | Flash 存储器编程模式 | | | |
| 储藏温度 | $T_{stg}$ | | | –40 ～ +150 | °C |

**注**　仅 48 引脚产品。

**注意事项**　任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因而，必须保证产品在不超过最大额定值的条件下使用。

**备注**　除非另外说明，复用功能引脚的特性与端口引脚的相同。

标准产品

**X1 振荡器特性**

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 振荡器 | 建议使用的电路 | 参数 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| 陶瓷振荡器 | $V_{SS}$ X1 X2<br>C1 C2 | X1 时钟振荡频率 ($f_X$)[注] | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 1.0 | | 20.0 | MHz |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 1.0 | | 10.0 | |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V | 1.0 | | 5.0 | |
| 晶体振荡器 | $V_{SS}$ X1 X2<br>C1 C2 | X1 时钟振荡频率 ($f_X$)[注] | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 1.0 | | 20.0 | MHz |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 1.0 | | 10.0 | |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V | 1.0 | | 5.0 | |

注　仅表示振荡器的特性。如需了解指令执行时间，可参见 **AC 特性** 。

注意事项 **1.** 在使用 **X1** 时钟时，上图中被虚线包围的部分的配线应按照如下配线方法配线，以防止连接线电容产生不利影响。

- 连接线越短越好。
- 连接线不应与其他信号线交叉。
- 流经的电流变化较大的信号线不要在振荡器周围布线。
- 要保持振荡器电容器的接地点电压与 $V_{SS}$ 相同。
- 不要将电容的地信号接入大电流地。
- 不要从振荡器获取信号。

**2.** 由于复位后 **CPU** 使用内部高速振荡时钟进行操作，因此用户要使用振荡稳定时间计数器的状态寄存器（**OSTC**）检测 **X1** 时钟振荡稳定时间。在充分评估了所使用的振荡器的振荡稳定时间后再确定 **OSTC** 和振荡稳定时间选择寄存器（**OSTS**）的振荡稳定时间。

标准产品

内部振荡器特性

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 振荡器 | 参数 | 条件 | | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| 8 MHz 内部振荡器 | 内部高速振荡时钟频率 ($f_{RH}$) 注 | RSTS = 1 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | 7.6 | 8.0 | 8.4 | MHz |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V | 7.6 | 8.0 | 10.4 | MHz |
| <R> | | RSTS = 0 | | 2.48 | 5 | 9.86 | MHz |
| 240 kHz 内部振荡器 | 内部低速振荡时钟频率 ($f_{RL}$) | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | 216 | 240 | 264 | kHz |
| <R> | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | 192 | 240 | 264 | kHz |

注 仅表示振荡器的特性。如需了解指令执行时间，可参见 **AC 特性** 。

备注 RSTS: 内部振荡模式寄存器 (RCM) 的第 7 位。

**XT1 荡器特性**

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 振荡器 | 建议使用的电路 | 参数 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| 晶体振荡器 | $V_{SS}$ XT2 XT1 Rd C4 C3 | XT1 时钟振荡频率($f_X$) 注 | | 32 | 32.768 | 35 | kHz |

注 仅表示振荡器的特性。如需了解指令执行时间，可参见 **AC 特性** 。

注意事项 **1.** 在使用 **XT1** 时钟时，上图中被虚线包围的部分的配线应按照如下配线方法配线，以防止连接线电容产生不利影响。

- 连接线越短越好。
- 连接线不应与其他信号线交叉。
- 流经的电流变化较大的信号线不要在振荡器周围布线。
- 要保持振荡器电容器的接地点电压与 $V_{SS}$ 相同。
- 不要将电容的地信号接入大电流地。
- 不要从振荡器获取信号。

**2.** **XT1** 振荡器是一个低振幅电路，用于降低功耗，但由于比 **X1** 振荡器更容易受到噪音干扰，因此在使用 **XT1** 时钟时应特别注意连线方式。

标准产品

<R>　推荐振荡器常数

**(1)　X1 振荡器: 陶瓷振荡器 ($T_A$ = –40 ～ +85°C) (1/2)**

| 厂商 | 编号 | SMD/ Lead | 频率 (MHz) | 推荐电路常数 | | 振荡电压范围 | |
|---|---|---|---|---|---|---|---|
| | | | | C1 (pF) | C2 (pF) | MIN. (V) | MAX. (V) |
| Murata Mfg. Co., Ltd. | CSTCC2M00G56-R0 | SMD | 2.00 | Internal (47) | Internal (47) | 1.8 | 5.5 |
| | CSTLS4M00G56-B0 | Lead | 4.00 | Internal (47) | Internal (47) | | |
| | CSTCR4M00G55-R0 | SMD | | Internal (39) | Internal (39) | | |
| | CSTLS4M19G56-B0 | Lead | 4.194 | Internal (47) | Internal (47) | | |
| | CSTCR4M19G55-R0 | SMD | | Internal (39) | Internal (39) | | |
| | CSTLS4M91G56-B0 | Lead | 4.915 | Internal (47) | Internal (47) | | |
| | CSTCR4M91G55-R0 | SMD | | Internal (39) | Internal (39) | | |
| | CSTLS5M00G56-B0 | Lead | 5.00 | Internal (47) | Internal (47) | 1.9 | |
| | CSTCR5M00G55-R0 | SMD | | Internal (39) | Internal (39) | 1.8 | |
| | CSTLS6M00G56-B0 | Lead | 6.00 | Internal (47) | Internal (47) | 2.4 | |
| | CSTCR6M00G55-R0 | SMD | | Internal (39) | Internal (39) | 1.8 | |
| | CSTLS8M00G56-B0 | Lead | 8.00 | Internal (47) | Internal (47) | 2.3 | |
| | CSTCE8M00G55-R0 | SMD | | Internal (33) | Internal (33) | 1.9 | |
| | CSTLS8M38G56-B0 | Lead | 8.388 | Internal (47) | Internal (47) | 2.3 | |
| | CSTCE8M38G55-R0 | SMD | | Internal (33) | Internal (33) | 1.9 | |
| | CSTLS10M0G56-B0 | Lead | 10.0 | Internal (47) | Internal (47) | 2.5 | |
| | CSTCE10M0G55-R0 | SMD | | Internal (33) | Internal (33) | 2.3 | |
| | CSTCE12M0G55-R0 | SMD | 12.0 | Internal (33) | Internal (33) | 2.3 | |
| | CSTCE16M0V53-R0 | SMD | 16.0 | Internal (15) | Internal (15) | 2.3 | |
| | CSTCE20M0V53-R0 | SMD | 20.0 | Internal (15) | Internal (15) | 2.6 | |
| Murata Mfg. Co., Ltd. (小电容产品 ) | CSTLS6M00G53-B0 | Lead | 6.00 | Internal (15) | Internal (15) | 1.8 | 5.5 |
| | CSTLS8M00G53-B0 | Lead | 8.00 | Internal (15) | Internal (15) | 1.8 | |
| | CSTLS8M38G53-B0 | Lead | 8.388 | Internal (15) | Internal (15) | 1.8 | |
| | CSTLS10M0G53-B0 | Lead | 10.0 | Internal (15) | Internal (15) | 1.8 | |
| | CSTCE12M0G52-R0 | SMD | 12.0 | Internal (10) | Internal (10) | 1.8 | |
| | CSTCE16M0V51-R0 | SMD | 16.0 | Internal (5) | Internal (5) | 1.8 | |
| | CSTCE20M0V51-R0 | SMD | 20.0 | Internal (5) | Internal (5) | 1.9 | |

注意事项　以上是根据振荡器厂家提供的在特定环境下的参考常数。如果在实际应用中需要优化振荡器特性，可以请厂家为应用电路做评价。振荡电压和频率仅仅包括振荡器特性。使用 **78K0/KC2** 以便内部操作条件包含在 **DC** 和 **AC** 特性中。

标准产品

**(1)　X1 振荡器: 陶瓷振荡器 ($T_A$ = –40 ～ +85°C) (2/2)**

| 厂商 | 编号 | SMD/ Lead | 频率 (MHz) | 推荐电路常数 | | 振荡电压范围 | |
|---|---|---|---|---|---|---|---|
| | | | | C1 (pF) | C2 (pF) | MIN. (V) | MAX. (V) |
| TDK Corporation | CCR4.0MUC8 | SMD | 4.00 | Internal (27) | Internal (27) | 1.8 | 5.5 |
| | FCR4.0MC5 | Lead | | Internal (30) | Internal (30) | | |
| | CCR8.0MXC8 | SMD | 8.00 | Internal (18) | Internal (30) | | |
| | FCR8.0MC5 | Lead | | Internal (20) | Internal (20) | | |

注意事项　以上是根据振荡器厂家提供的在特定环境下的参考常数。如果在实际应用中需要优化振荡器特性，可以请厂家为应用电路做评价。振荡电压和频率仅仅包括振荡器特性。使用 **78K0/KC2** 以便内部操作条件包含在 **DC** 和 **AC** 特性中。

**(2)　XT1 振荡器: 晶体振荡器 ($T_A$ = –40 ～ +85°C)**

| 厂商 | 编号 | SMD/ Lead | 频率 (MHz) | 推荐电路常数 | 振荡电压范围 | | | | | | 振荡电压范围 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | $V_{DD}$ = 3.3 V | | | $V_{DD}$ = 5.0 V | | | MIN. (V) | MAX. (V) |
| | | | | | C3 (pF) | C4 (pF) | Rd (k ) | C3 (pF) | C4 (pF) | Rd (k ) | | |
| Seiko Instruments Inc. | VT-200 | Lead | 32.768 | 6.0 | 4 | 3 | 100 | 6 | 5 | 100 | 1.8 | 5.5 |
| | | | | 12.5 | 15 | 15 | 100 | 18 | 15 | 100 | | |

注意事项　以上是根据振荡器厂家提供的在特定环境下的参考常数。如果在实际应用中需要优化振荡器特性，可以请厂家为应用电路做评价。振荡电压和频率仅仅包括振荡器特性。使用 **78K0/KC2** 以便内部操作条件包含在 **DC** 和 **AC** 特性中。

标准产品

**DC 特性 (1/5)**

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 参数 | 符号 | 条件 | | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| 输出电流，高[注1] | $I_{OH1}$ | P00, P01, P10 ～ P17, P30 ～ P33, P40, P41, P70 ～ P73, P74[注2], P75[注2], P120, P130[注2], P140[注2]的每个引脚 | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | | –3.0 | mA |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | | –2.5 | mA |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | | –1.0 | mA |
| | | 引脚总和[注4] P00, P01, P40, P41, P120, P130[注2], P140[注2] | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | | –20.0 | mA |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | | –10.0 | mA |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | | –5.0 | mA |
| | | 引脚总和[注4] P10 ～ P17, P30 ～ P33, P70 ～ P73, P74[注2], P75[注2] | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | | –30.0 | mA |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | | –19.0 | mA |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | | –10.0 | mA |
| | | 所有引脚总和[注4] | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | | –50.0 | mA |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | | –29.0 | mA |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | | –15.0 | mA |
| | $I_{OH2}$ | P20 ～ P27 的每个引脚 | $AV_{REF}$ = $V_{DD}$ | | | –0.1 | mA |
| | | P121 ～ P124 的每个引脚 | | | | –0.1 | mA |
| 输出电流，低[注3] | $I_{OL1}$ | P00, P01, P10 ～ P17, P30 ～ P33, P40, P41, P70 ～ P73, P74[注2], P75[注2], P120, P130[注2], P140[注2]的每个引脚 | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | | 8.5 | mA |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | | 5.0 | mA |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | | 2.0 | mA |
| | | P60 ～ P63 的每个引脚 | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | | 15.0 | mA |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | | 5.0 | mA |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | | 2.0 | mA |
| | | 引脚总和[注4] P00, P01, P40, P41, P120, P130[注2], P140[注2] | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | | 20.0 | mA |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | | 15.0 | mA |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | | 9.0 | mA |
| | | 引脚总和[注4] P10 ～ P17, P30 ～ P33, P60 ～ P63, P70 ～ P73, P74[注2], P75[注2] | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | | 45.0 | mA |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | | 35.0 | mA |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | | 20.0 | mA |
| | | 所有引脚总和[注4] | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | | 65.0 | mA |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | | 50.0 | mA |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | | 29.0 | mA |

**注** **1.** 即使电流从 $V_{DD}$ 流向输出引脚，设备操作时的电流值也可以得到保证。

**2.** 仅 48 引脚产品。

**3.** 即使电流从输出引脚流向 GND，设备操作时的电流值也可以得到保证。

**4.** 占空比=70%时 (输出电流的时间为 0.7 × t，不输出电流的时间为 0.3 × t，其中 t 为指定时间)的规定。可通过以下表达式计算占空比不等于 70%时的引脚总的输出电流。

- 当 $I_{OH}$ 的占空比为 n%: 引脚总的输出电流 = ($I_{OH}$ × 0.7)/(n × 0.01)

<举例>　当占空比为 50%, $I_{OH}$ = 20.0 mA

引脚总的输出电流 = (20.0 × 0.7)/(50 × 0.01) = 28.0 mA

但是，允许流入每个引脚的电流不会随占空比而变化。高于最大额定值的电流不能流入引脚。

**备注**　除非另外说明，复用功能引脚的特性与端口引脚的相同。

标准产品

**DC 特性 (2/5)**

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 参数 | 符号 | 条件 | | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| 输出电流，低[注1] | $I_{OL2}$ | P20 ～ P27 的每个引脚 | $AV_{REF}$ = $V_{DD}$ | | | 0.4 | mA |
| | | P121 ～ P124 的每个引脚 | | | | 0.4 | mA |
| <R> 输入电压，高 | $V_{IH1}$ | P12, P13, P15, P40, P41, P121 ～ P124 | | 0.7$V_{DD}$ | | $V_{DD}$ | V |
| <R> | $V_{IH2}$ | P00, P01, P10, P11, P14, P16, P17, P30 ～ P33, P70 ～ P73, P74[注2], P75[注2], P120, P140[注2], $\overline{\text{RESET}}$ | | 0.8$V_{DD}$ | | $V_{DD}$ | V |
| | $V_{IH3}$ | P20 ～ P27 | $AV_{REF}$ = $V_{DD}$ | 0.7$AV_{REF}$ | | $AV_{REF}$ | V |
| <R> | $V_{IH4}$ | P60 ～ P63 | | 0.7$V_{DD}$ | | 6.0 | V |
| <R> <R> 输入电压，低 | $V_{IL1}$ | P12, P13, P15, P40, P41, P60 ～ P63, P121 ～ P124 | | 0 | | 0.3$V_{DD}$ | V |
| | $V_{IL2}$ | P00, P01, P10, P11, P14, P16, P17, P30 ～ P33, P70 ～ P73, P74[注2], P75[注2], P120, P140[注2], $\overline{\text{RESET}}$ | | 0 | | 0.2$V_{DD}$ | V |
| | $V_{IL3}$ | P20 ～ P27 | $AV_{REF}$ = $V_{DD}$ | 0 | | 0.3$AV_{REF}$ | V |
| 输出电压，高 | $V_{OH1}$ | P00, P01, P10 ～ P17, P30 ～ P33, P40, P41, P70 ～ P73, P74[注2], P75[注2], P120, P130[注2], P140[注2] | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $I_{OH1}$ = –3.0 mA | $V_{DD}$ – 0.7 | | | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $I_{OH1}$ = –2.5 mA | $V_{DD}$ – 0.5 | | | V |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V, $I_{OH1}$ = –1.0 mA | $V_{DD}$ – 0.5 | | | V |
| | $V_{OH2}$ | P20 ～ P27 | $AV_{REF}$ = $V_{DD}$, $I_{OH2}$ = –100 $\mu$A | $V_{DD}$ – 0.5 | | | V |
| | | P121 ～ P124 | $I_{OH2}$ = –100 $\mu$A | $V_{DD}$ – 0.5 | | | V |

**注**　**1.** 即使电流从输出引脚流向 GND，设备操作时的电流值也可以得到保证。

　　**2.** 仅 48 引脚产品。

**备注**　除非另外说明，复用功能引脚的特性与端口引脚的相同。

标准产品

**DC 特性 (3/5)**

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 参数 | 符号 | 条件 | | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| 输出电压，低 | $V_{OL1}$ | P00, P01, P10 ～ P17, P30 ～ P33, P40, P41, P70 ～ P73, P74[注], P75[注], P120, P130[注], P140[注] | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $I_{OL1}$ = 8.5 mA | | | 0.7 | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $I_{OL1}$ = 5.0 mA | | | 0.7 | V |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V, $I_{OH1}$ = 2.0 mA | | | 0.5 | V |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V, $I_{OL1}$ = 0.5 mA | | | 0.4 | V |
| | $V_{OL2}$ | P20 ～ P27 | $AV_{REF}$ = $V_{DD}$, $I_{OL2}$ = 0.4 mA | | | 0.4 | V |
| | | P121 ～ P124 | $I_{OL2}$ = 0.4 mA | | | 0.4 | V |
| | $V_{OL3}$ | P60 ～ P63 | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $I_{OL1}$ = 15.0 mA | | | 2.0 | V |
| | | | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $I_{OL1}$ = 5.0 mA | | | 0.4 | V |
| <R> | | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $I_{OL1}$ = 5.0 mA | | | 0.6 | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $I_{OL1}$ = 3.0 mA | | | 0.4 | V |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V, $I_{OL1}$ = 2.0 mA | | | 0.4 | V |
| 输入泄漏电流，高 | $I_{LIH1}$ | P00, P01, P10 ～ P17, P30 ～ P33, P40, P41, P60 ～ P63, P70 ～ P73, P74[注], P75[注], P120, P140[注] | $V_I$ = $V_{DD}$ | | | 1 | $\mu$A |
| | $I_{LIH2}$ | P20 ～ P27 | $V_I$ = $AV_{REF}$ = $V_{DD}$ | | | 1 | $\mu$A |
| | $I_{LIH3}$ | P121 ～ 124 (X1, X2, XT1, XT2) | $V_I$ = $V_{DD}$ I/O 端口模式 | | | 1 | $\mu$A |
| | | | $V_I$ = $V_{DD}$ OSC 模式 | | | 20 | $\mu$A |
| 输入泄漏电流，低 | $I_{LIL1}$ | P00, P01, P10 ～ P17, P30 ～ P33, P40, P41, P60 ～ P63, P70 ～ P73, P74[注], P75[注], P120, P140[注] | $V_I$ = $V_{SS}$ | | | –1 | $\mu$A |
| | $I_{LIL2}$ | P20 ～ P27 | $V_I$ = $V_{SS}$, $AV_{REF}$ = $V_{DD}$ | | | –1 | $\mu$A |
| | $I_{LIL3}$ | P121 ～ 124 (X1, X2, XT1, XT2) | $V_I$ = $V_{SS}$ I/O 端口模式 | | | –1 | $\mu$A |
| | | | $V_I$ = $V_{SS}$ OSC 模式 | | | –20 | $\mu$A |
| 上拉电阻 | $R_U$ | $V_I$ = $V_{DD}$ | | 10 | 20 | 100 | kΩ |
| FLMD0 电压 | $V_{IL}$ | 正常操作模式 | | 0 | | 0.2$V_{DD}$ | V |
| | $V_{IH}$ | 自编程模式 | | 0.8$V_{DD}$ | | $V_{DD}$ | V |
| <R> 外部时钟输入电压 | $V_{IL}$ | P122, P124 | 外部时钟输入模式 | 0 | | 0.3$V_{DD}$ | V |
| | $V_{IH}$ | | | 0.7$V_{DD}$ | | $V_{DD}$ | V |

**注**　　仅 48 引脚产品。

**备注**　　除非另外说明，复用功能引脚的特性与端口引脚的相同。

标准产品

**DC 特性 (4/5)**

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 参数 | 符号 | 条件 | | | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|---|
| <R> 电源电流 | $I_{DD1}$[注1] | 运行模式 | $f_{XH}$ = 20 MHz[注2], $V_{DD}$ = 5.0 V | 方波输入 | | 3.2 | 5.5 | mA |
| | | | | 振荡器连接 | | 4.5 | 6.9 | mA |
| | | | $f_{XH}$ = 10 MHz[注2, 3], $V_{DD}$ = 5.0 V | 方波输入 | | 1.6 | 2.8 | mA |
| | | | | 振荡器连接 | | 2.3 | 3.9 | mA |
| | | | $f_{XH}$ = 10 MHz[注2, 3], $V_{DD}$ = 3.0 V | 方波输入 | | 1.5 | 2.7 | mA |
| | | | | 振荡器连接 | | 2.2 | 3.2 | mA |
| | | | $f_{XH}$ = 5 MHz[注2, 3], $V_{DD}$ = 3.0 V | 方波输入 | | 0.9 | 1.6 | mA |
| | | | | 振荡器连接 | | 1.3 | 2.0 | mA |
| | | | $f_{XH}$ = 5 MHz[注2, 3], $V_{DD}$ = 2.0 V | 方波输入 | | 0.7 | 1.4 | mA |
| | | | | 振荡器连接 | | 1.0 | 1.6 | mA |
| | | | $f_{RH}$ = 8 MHz, $V_{DD}$ = 5.0 V | | | 1.4 | 2.5 | mA |
| | | | $f_{SUB}$ = 32.768 kHz[注2, 4], $V_{DD}$ = 5.0 V | 方波输入 | | 6 | 25 | µA |
| | | | | 振荡器连接 | | 15 | 30 | µA |
| | $I_{DD2}$[注5] | HALT 模式 | $f_{XH}$ = 20 MHz[注2], $V_{DD}$ = 5.0 V | 方波输入 | | 0.8 | 2.6 | mA |
| | | | | 振荡器连接 | | 2.0 | 4.4 | mA |
| | | | $f_{XH}$ = 10 MHz[注2, 3], $V_{DD}$ = 5.0 V | 方波输入 | | 0.4 | 1.3 | mA |
| | | | | 振荡器连接 | | 1.0 | 2.4 | mA |
| | | | $f_{XH}$ = 5 MHz[注2, 3], $V_{DD}$ = 3.0 V | 方波输入 | | 0.2 | 0.65 | mA |
| | | | | 振荡器连接 | | 0.5 | 1.1 | mA |
| | | | $f_{RH}$ = 8 MHz, $V_{DD}$ = 5.0 V | | | 0.4 | 1.2 | mA |
| | | | $f_{SUB}$ = 32.768 kHz[注2, 4], $V_{DD}$ = 5.0 V | 方波输入 | | 3.0 | 22 | µA |
| | | | | 振荡器连接 | | 12 | 25 | µA |
| | $I_{DD3}$[注5] | STOP 模式 | $V_{DD}$ = 5.0 V | | | 1 | 20 | µA |
| | | | $V_{DD}$ = 5.0 V, $T_A$ = –40 ～ +70°C | | | 1 | 10 | µA |

<R> 注 **1.** 所有内部电源($V_{DD}$, $AV_{REF}$)电流，包括外围操作电流（而不包括电流流入端口的上拉电阻，A/D 转换器的部分）。

<R> **2.** 不包括内部高速时钟($f_{RH}$)的操作电流。

**3.** 当 AMPH (时钟操作模式选择寄存器 (OSCCTL) 的第 0 位 ) = 0 。

**4.** 当停止主系统时钟时。

<R> **5.** 所有内部电源($V_{DD}$, $AV_{REF}$)电流，包括外围操作电流（而不包括电流流入端口的上拉电阻，A/D 转换器，看门狗定时器，和 LVI 电路的部分）。

备注 **1.** $f_{XH}$: 高速系统时钟频率 (X1 时钟振荡频率或外部主系统时钟频率)

**2.** $f_{RH}$: 内部高速振荡时钟频率

**3.** $f_{SUB}$: 副系统时钟频率 (XT1 时钟振荡频率或外部副系统时钟频率)

标准产品

**DC 特性 (5/5)**

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 参数 | 符号 | 条件 | | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| A/D 转换器工作电流 | $I_{ADC}$ 注 1 | 在最大转换速度时 | 2.3 V ≤ $AV_{REF}$ ≤ $V_{DD}$ | | 0.86 | 1.9 | mA |
| <R> | | 非转换时 (比较器操作) 注 2 | | | 0.6 | 1.2 | mA |
| 看门狗定时器工作电流 | $I_{WDT}$ 注 3 | 在 240 kHz 内部低速振荡时钟下工作 | | | 5 | 10 | $\mu$A |
| <R> LVI 工作电流 | $I_{LVI}$ 注 4 | | | | 9 | 18 | $\mu$A |

注 **1.** 电流仅流入 A/D 转换器中。在操作或 HALT 模式下当 A/D 转换器工作时 78K0/KC2 电流值为 $I_{DD1}$ 或 $I_{DD2}$ 和 $I_{ADC}$ 之和。

<R> **2.** 当 ADCE (A/D 转换器模式寄存器 (ADM) 的第 0 位 ) = 1 并且 ADCS (ADM 的第 7 位) = 0 时的电流。

**3.** 电流流入看门狗定时器中。在 HALT 或 STOP 模式下当看门狗定时器工作时 78K0/KC2 电流值为 $I_{DD2}$ 或 $I_{DD3}$ 和 $I_{WDT}$ 之和。

**4.** 电流流入 LVI 电路。在 HALT 或 STOP 模式下当 LVI 电路工作时 78K0/KC2 电流值为 $I_{DD2}$ 或 $I_{DD3}$ 和 $I_{LVI}$ 之和。

标准产品

## AC 特性

### (1) 基本操作

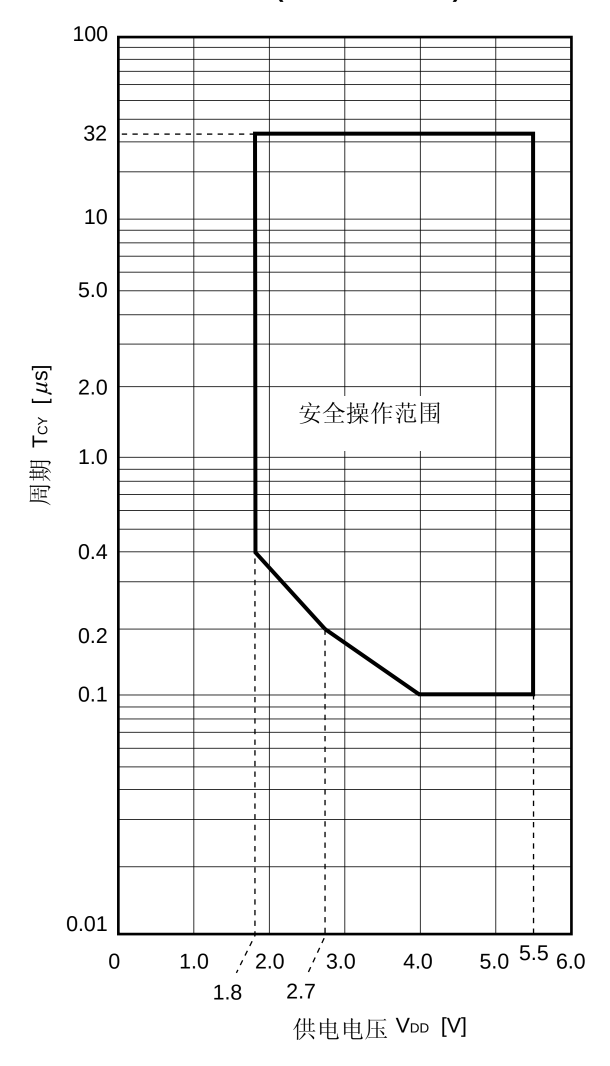
**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 参数 | 符号 | 条件 | | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| 指令周期（最小指令执行时间） | $T_{CY}$ | 主系统时钟 ($f_{XP}$) 操作 | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0.1 | | 32 | μs |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0.2 | | 32 | μs |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V | 0.4[注 1] | | 32 | μs |
| | | 副系统时钟 ($f_{SUB}$) 操作 | | 114 | 122 | 125 | μs |
| 外部主系统时钟频率 | $f_{EXCLK}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.0 | | 20.0 | MHz |
| | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | 1.0 | | 10.0 | MHz |
| | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | 1.0 | | 5.0 | MHz |
| 外部主系统时钟输入高电平宽度，低电平宽度 | $t_{EXCLKH}$, $t_{EXCLKL}$ | | | $(1/f_{EXCLK} \times 1/2) - 1$ | | | ns |
| 外部副系统时钟频率 | $f_{EXCLKS}$ | | | 32 | 32.768 | 35 | kHz |
| 外部副系统时钟输入高电平宽度，低电平宽度 | $t_{EXCLKSH}$, $t_{EXCLKSL}$ | | | $(1/f_{EXCLKS} \times 1/2) - 5$ | | | ns |
| TI000, TI010 输入高电平宽度，低电平宽度 | $t_{TIH0}$, $t_{TIL0}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | $2/f_{sam}$ + 0.1[注 2] | | | μs |
| | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | $2/f_{sam}$ + 0.2[注 2] | | | μs |
| <R> | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | $2/f_{sam}$ + 0.5[注 2] | | | μs |
| TI50, TI51 输入频率 | $f_{TI5}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | | | 10 | MHz |
| | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | | | 10 | MHz |
| | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | | | 5 | MHz |
| TI50, TI51 输入高电平宽度，低电平宽度 | $t_{TIH5}$, $t_{TIL5}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | 50 | | | ns |
| | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | 50 | | | ns |
| | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | 100 | | | ns |
| 中断输入高电平宽度，低电平宽度 | $t_{INTH}$, $t_{INTL}$ | | | 1 | | | μs |
| 按键中断输入低电平宽度 | $t_{KR}$ | | | 250 | | | ns |
| $\overline{RESET}$ 低电平宽度 | $t_{RSL}$ | | | 10 | | | μs |

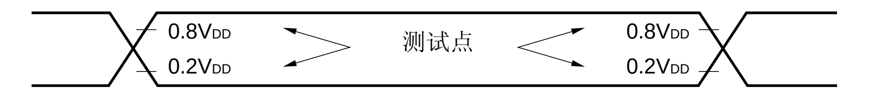
注 **1.** 当使用 8 MHz 内部振荡器时为 0.38 μs。

**2.** 使用预分频模式寄存器 00（PRM00）的第 0 和 1 位（PRM000、PRM001）可以选择 $f_{sam}$ = $f_{PRS}$, $f_{PRS}/4$, $f_{PRS}/256$。注意，当选择 TI000 的有效沿作为计数时钟时，$f_{sam}$ = $f_{PRS}$ 。

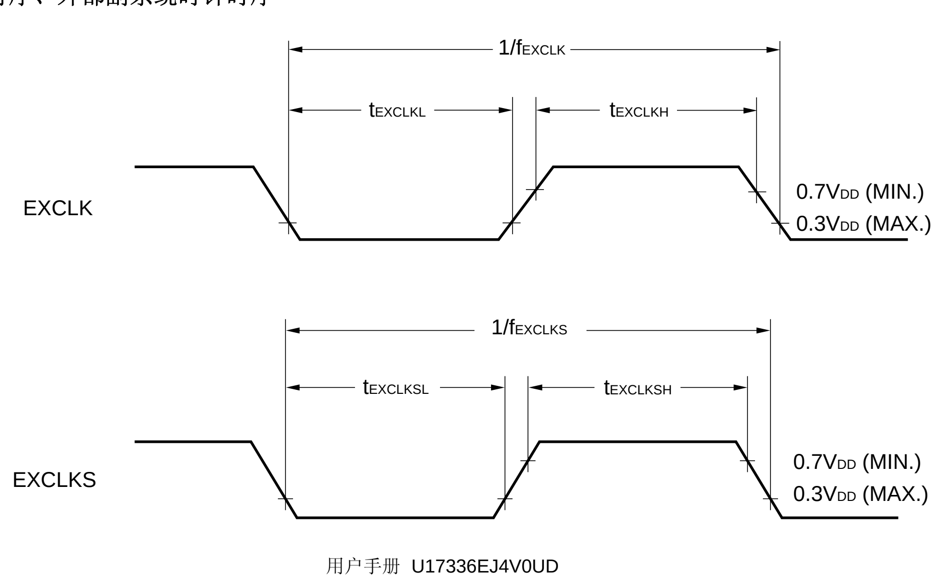
**$T_{CY}$ vs. $V_{DD}$ (主系统时钟操作)**



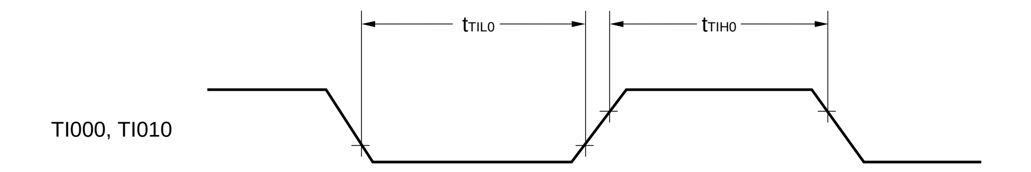

**AC** 时序测试点 **(**外部主系统时钟和外部副系统时钟除外**)**

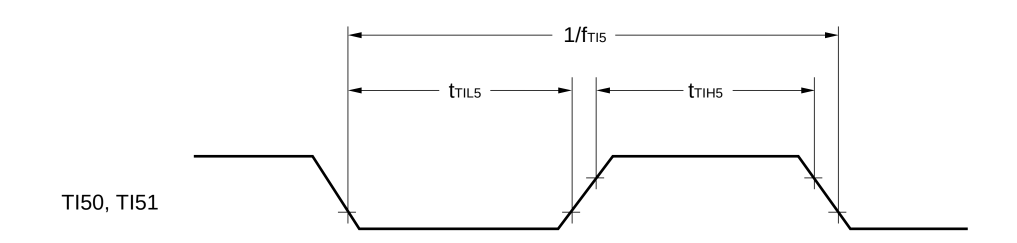



外部主系统时钟时序、外部副系统时钟时序

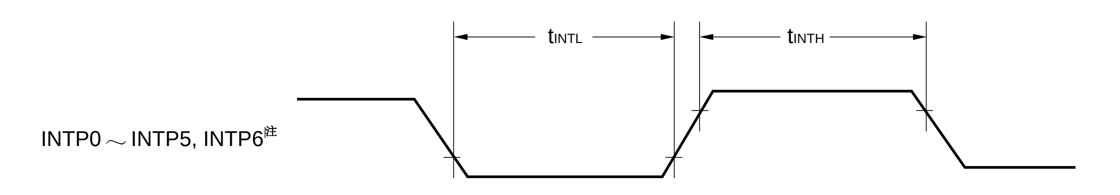


**TI** 时序



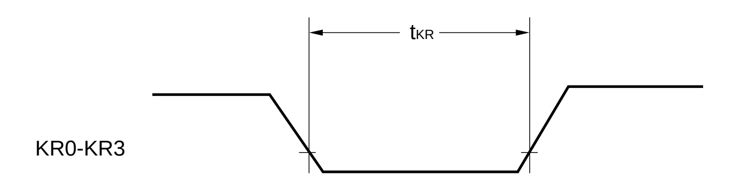




中断请求输入时序

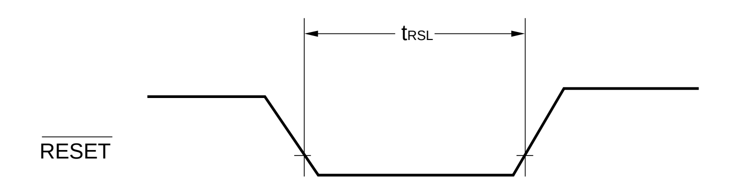



注　　仅 48 引脚产品。

按键中断输入时序




$\overline{\textbf{RESET}}$ 输入时序



标准产品

**(2)** 串行接口

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

**(a) UART6 (专用波特率发生器输出)**

| 参数 | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|
| <R> 传输速率 | | | | | 625 | kbps |

**(b) UART0 (专用波特率发生器输出)**

| 参数 | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|
| <R> 传输速率 | | | | | 625 | kbps |

**(c) IIC0**

| 参数 | 符号 | 标准模式 | | 高速模式 | | 单位 |
|---|---|---|---|---|---|---|
| | | MIN. | MAX. | MIN. | MAX. | |
| SCL0 时钟频率 | $f_{SCL}$ | 0 | 100 | 0 | 400 | kHz |
| 启动/重新启动条件的建立时间[注 1] | $t_{SU:STA}$ | 4.8 | – | 0.7 | – | μs |
| 保持时间 | $t_{HD:STA}$ | 4.1 | – | 0.7 | – | μs |
| 当 SCL0 = “L”时的保持时间 | $t_{LOW}$ | 5.0 | – | 1.25 | – | μs |
| 当 SCL0 = “H”时的保持时间 | $t_{HIGH}$ | 5.0 | – | 1.25 | – | μs |
| 数据建立时间 (接收) | $t_{SU:DAT}$ | 0 | – | 0 | – | μs |
| 数据保持时间 (发送)[注 2] | $t_{HD:DAT}$ | 0.47 | 4.0 | 0.23 | 1.00 | μs |

注 **1.** 当检测到启动/重新启动条件时在经历了该周期时间后产生第一个时钟脉冲。

**2.** $t_{HD:DAT}$ 的最大值 (MAX.) 是在正常传送期间，在 $\overline{ACK}$ (应答)时序中插入一个等待状态。

标准产品

**(d) CSI10 (主设备模式, $\overline{\text{SCK10}}$... 内部时钟输出)**

| 参数 | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|
| $\overline{\text{SCK10}}$ 周期时间 | $t_{KCY1}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 100 | | | ns |
| | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 200 | | | ns |
| | | 1.8 V ≤ $V_{DD}$ < 2.7 V | 400 | | | ns |
| $\overline{\text{SCK10}}$ 高/低电平宽度 | $t_{KH1}$, $t_{KL1}$ | | $t_{KCY1}/2 - 10$[注1] | | | ns |
| SI10 建立时间 (到 $\overline{\text{SCK10}}$↑) | $t_{SIK1}$ | | 30 | | | ns |
| SI10 保持时间 (从 $\overline{\text{SCK10}}$↑) | $t_{KSI1}$ | | 30 | | | ns |
| 从 $\overline{\text{SCK10}}$↓到 SO10 输出的延迟时间 | $t_{KSO1}$ | C = 50 pF[注2] | | | 40 | ns |

注 **1.** 使用高速系统时钟 ($f_{XH}$) 时的取值

**2.** C 是 $\overline{\text{SCK10}}$ 和 SO10 输出线的负载电容。

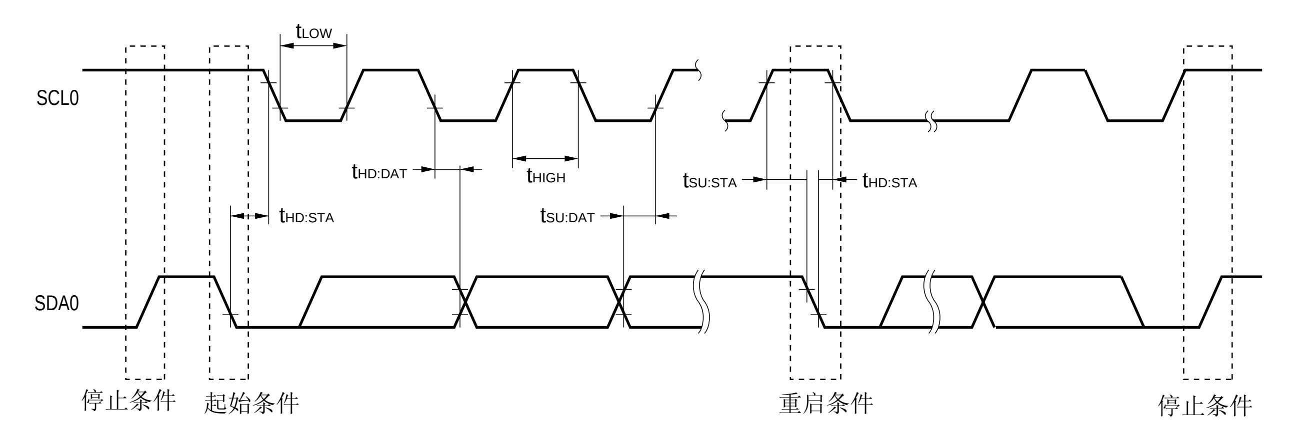
**(e) CSI10 (从设备模式, SCK1n... 外部时钟输入)**

| 参数 | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|
| $\overline{\text{SCK10}}$ 周期 | $t_{KCY2}$ | | 400 | | | ns |
| $\overline{\text{SCK10}}$ 高/低电平宽度 | $t_{KH2}$, $t_{KL2}$ | | $t_{KCY2}/2$ | | | ns |
| SI10 建立时间 (至 $\overline{\text{SCK10}}$↑) | $t_{SIK2}$ | | 80 | | | ns |
| SI10 保持时间 (从 $\overline{\text{SCK10}}$↑) | $t_{KSI2}$ | | 50 | | | ns |
| 从 $\overline{\text{SCK10}}$↓ 到 SO10 输出的延迟时间 | $t_{KSO2}$ | C = 50 pF[注] | | | 120 | ns |

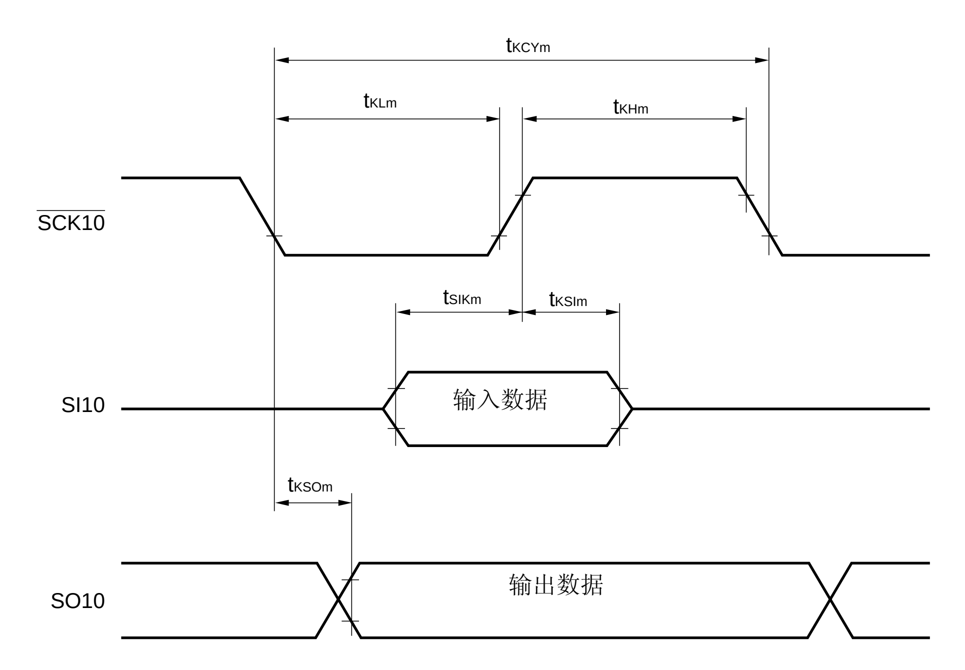
注 C 是 SO10 输出线的负载电容。

串行传输时序

**IIC0:**




**CSI10:**




备注　　m = 1, 2

**A/D 转换器特性**

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, 2.3 V ≤ $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 参数 | 符号 | 条件 | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| 分辨率 | $R_{ES}$ | | | | 10 | bit |
| 总误差注 [1, 2] | $A_{INL}$ | 4.0 V ≤ $AV_{REF}$ ≤ 5.5 V | | | ±0.4 | %FSR |
| | | 2.7 V ≤ $AV_{REF}$ < 4.0 V | | | ±0.6 | %FSR |
| | | 2.3 V ≤ $AV_{REF}$ < 2.7 V | | | ±1.2 | %FSR |
| 转换时间 | $t_{CONV}$ | 4.0 V ≤ $AV_{REF}$ ≤ 5.5 V | 6.1 | | 36.7 | μs |
| | | 2.7 V ≤ $AV_{REF}$ < 4.0 V | 12.2 | | 36.7 | μs |
| | | 2.3 V ≤ $AV_{REF}$ < 2.7 V | 27 | | 66.6 | μs |
| 零度误差注 [1, 2] | $E_{ZS}$ | 4.0 V ≤ $AV_{REF}$ ≤ 5.5 V | | | ±0.4 | %FSR |
| | | 2.7 V ≤ $AV_{REF}$ < 4.0 V | | | ±0.6 | %FSR |
| | | 2.3 V ≤ $AV_{REF}$ < 2.7 V | | | ±0.6 | %FSR |
| 满度误差注 [1, 2] | $E_{FS}$ | 4.0 V ≤ $AV_{REF}$ ≤ 5.5 V | | | ±0.4 | %FSR |
| | | 2.7 V ≤ $AV_{REF}$ < 4.0 V | | | ±0.6 | %FSR |
| | | 2.3 V ≤ $AV_{REF}$ < 2.7 V | | | ±0.6 | %FSR |
| 积分非线性误差注 [1] | $I_{LE}$ | 4.0 V ≤ $AV_{REF}$ ≤ 5.5 V | | | ±2.5 | LSB |
| | | 2.7 V ≤ $AV_{REF}$ < 4.0 V | | | ±4.5 | LSB |
| | | 2.3 V ≤ $AV_{REF}$ < 2.7 V | | | ±6.5 | LSB |
| 微分非线性误差注 [1] | $D_{LE}$ | 4.0 V ≤ $AV_{REF}$ ≤ 5.5 V | | | ±1.5 | LSB |
| | | 2.7 V ≤ $AV_{REF}$ < 4.0 V | | | ±2.0 | LSB |
| | | 2.3 V ≤ $AV_{REF}$ < 2.7 V | | | ±2.0 | LSB |
| 模拟输入电压 | $V_{AIN}$ | | $AV_{SS}$ | | $AV_{REF}$ | V |

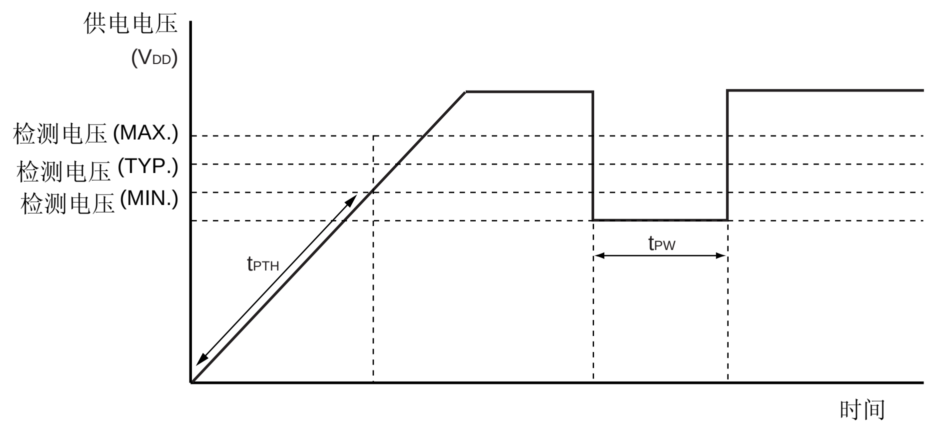
注　**1.** 不包括量化误差 (±1/2 LSB)

　　**2.** 该值表示满度的百分比 (%FSR)

**1.59 V POC 电路特性 ($T_A$ = –40 ～ +85°C, $V_{SS}$ = 0 V)**

| 参数 | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|
| 检测电压 | $V_{POC}$ | | 1.44 | 1.59 | 1.74 | V |
| <R> 供电电压的上升斜率 | $t_{PTH}$ | $V_{DD}$: 0 V → 变到 $V_{POC}$ 的指定电压 | 0.5 | | | V/ms |
| 最小脉冲宽度 | $t_{PW}$ | | 200 | | | μs |

**POC 电路时序**



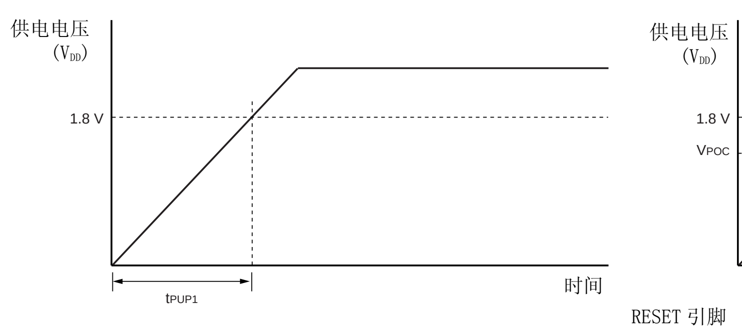
标准产品

### 供电电压上升时间 ($T_A$ = –40 ～ +85°C, $V_{SS}$ = 0 V)

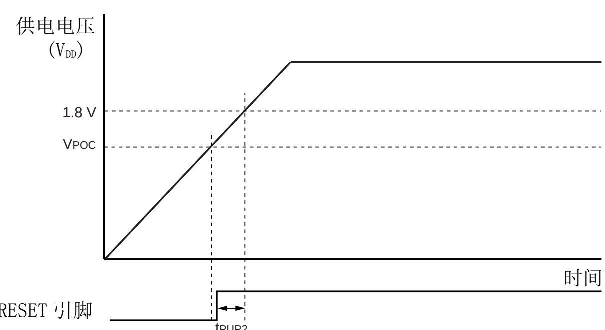
| 参数 | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|
| 上升至 1.8 V ($V_{DD}$ (MIN.))所需的最长时间 ($V_{DD}$: 0 V → 1.8 V) | $t_{PUP1}$ | 当没有使用 $\overline{RESET}$ 输入时，POCMODE (选项字节) = 0 | | | 3.6 | ms |
| 上升至 1.8 V ($V_{DD}$ (MIN.))所需的最长时间 (释放 $\overline{RESET}$ 输入 → $V_{DD}$: 1.8 V) | $t_{PUP2}$ | 当使用 $\overline{RESET}$ 输入时，POCMODE (选项字节) = 0 | | | 1.9 | ms |

供电电压上升时间时序

• 当没有使用 $\overline{RESET}$ 引脚输入时




•当使用 $\overline{RESET}$ 引脚输入时




### 2.7 V POC 电路特性 ($T_A$ = –40 ～ +85°C, $V_{SS}$ = 0 V)

| 参数 | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|
| 供电电压应用中的检测电压 | $V_{DDPOC}$ | POCMODE (选项字节) = 1 | 2.50 | 2.70 | 2.90 | V |

标准产品

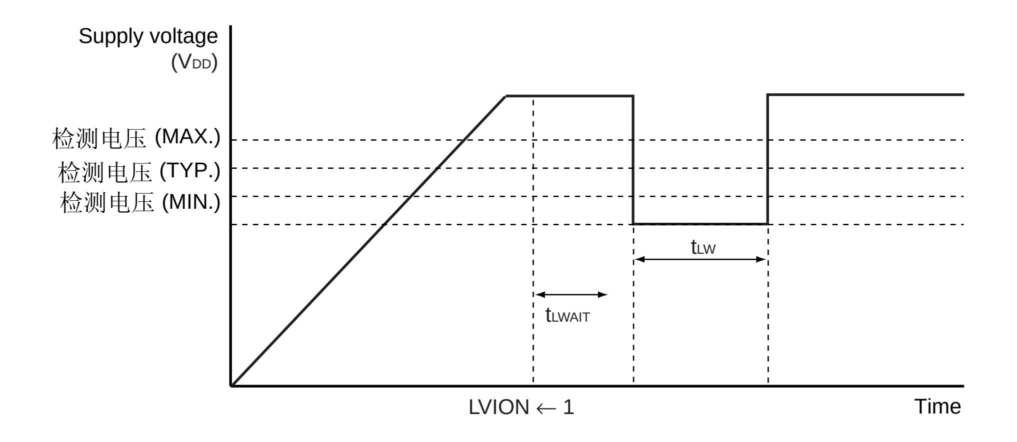
**LVI 电路特性 ($T_A$ = –40 ～ +85°C, $V_{POC} \leq V_{DD} \leq 5.5$ V, $AV_{REF} \leq V_{DD}$, $V_{SS}$ = 0 V)**

| 参数 | | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| 检测电压 | 供电电压电平 | $V_{LVI0}$ | | 4.14 | 4.24 | 4.34 | V |
| | | $V_{LVI1}$ | | 3.99 | 4.09 | 4.19 | V |
| | | $V_{LVI2}$ | | 3.83 | 3.93 | 4.03 | V |
| | | $V_{LVI3}$ | | 3.68 | 3.78 | 3.88 | V |
| | | $V_{LVI4}$ | | 3.52 | 3.62 | 3.72 | V |
| | | $V_{LVI5}$ | | 3.37 | 3.47 | 3.57 | V |
| | | $V_{LVI6}$ | | 3.22 | 3.32 | 3.42 | V |
| | | $V_{LVI7}$ | | 3.06 | 3.16 | 3.26 | V |
| | | $V_{LVI8}$ | | 2.91 | 3.01 | 3.11 | V |
| | | $V_{LVI9}$ | | 2.75 | 2.85 | 2.95 | V |
| | | $V_{LVI10}$ | | 2.60 | 2.70 | 2.80 | V |
| | | $V_{LVI11}$ | | 2.45 | 2.55 | 2.65 | V |
| | | $V_{LVI12}$ | | 2.29 | 2.39 | 2.49 | V |
| | | $V_{LVI13}$ | | 2.14 | 2.24 | 2.34 | V |
| | | $V_{LVI14}$ | | 1.98 | 2.08 | 2.18 | V |
| | | $V_{LVI15}$ | | 1.83 | 1.93 | 2.03 | V |
| <R> | 外部输入引脚[注 1] | EXLVI | EXLVI < $V_{DD}$, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V | 1.11 | 1.21 | 1.31 | V |
| 最短脉冲宽度 | | $t_{LW}$ | | 200 | | | μs |
| 操作稳定等待时间[注 2] | | $t_{LWAIT}$ | | | | 10 | μs |

注　**1.** 使用 EXLVI/P120/INTP0 引脚
　　**2.** 从设置低电压检测寄存器(LVIM)的第 7 位(LVION)=1 到操作稳定所需的时间。

备注　$V_{LVI(n-1)} > V_{LVIn}$: n = 1 ～ 15

LVI 电路时序

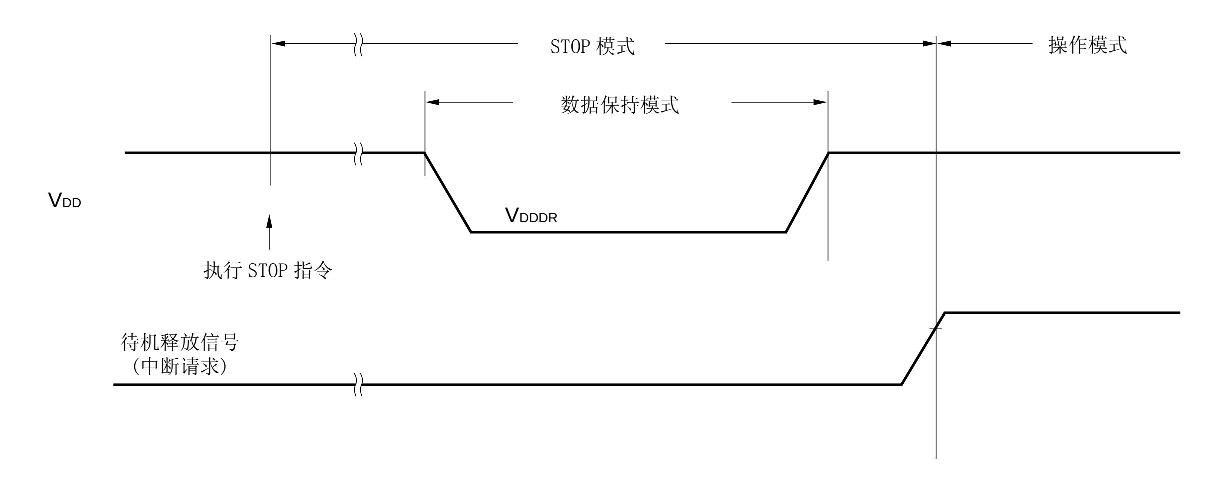


标准产品

**数据存储器在 STOP 模式下低供电电压时的数据保持特性 ($T_A$ = –40 ～ +85°C)**

| 参数 | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|
| 数据保持供电电压 | $V_{DDDR}$ | | 1.44[注] | | 5.5 | V |

注　该值根据 POC 检测电压而变化。当电压降低时，一直保持该数据直至 POC 复位有效，但当 POC 复位有效时数据不能保持。




<R>

**Flash 存储器编程特性**
**($T_A$ = –40 ～ +85°C, 2.7 V ≤ $V_{DD}$ ≤ 5.5 V, $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

- 基本特性

| 参数 | | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| $V_{DD}$ 供电电流 | | $I_{DD}$ | $f_{XP}$ = 10 MHz (TYP.), 20 MHz (MAX.) | | 4.5 | 11.0 | mA |
| 擦除时间[注 1, 2] | 所有 block | $T_{eraca}$ | | | 20 | 200 | ms |
| | Block 单元 | $T_{erasa}$ | | | 20 | 200 | ms |
| 写时间(以 8 位为单位)[注 1] | | $T_{wrwa}$ | | | 10 | 100 | μs |
| 每片重写次数 | | $C_{erwr}$ | 保持时间: 10 年<br>擦除 1 次 +擦除后写 1 次= 1 次重写[注 3] | 100 | | | 次数 |

<R>

注　**1.** flash 存储器的特性。当使用专用 flash 存储器编程器，PG-FP4 时，会应用这些特性，并且自编程下的重写次数，参看表 **25-12** 和 **25-13**。
　　**2.** 不包括擦除前的预写时间和擦除验证时间(写回时间)。
　　**3.** 在出货后第一次对产品进行写操作时，“擦除 → 写”和 “只写”作为一次重写。

**备注**　**1.** $f_{XP}$: 主系统时钟振荡频率

<R>

　　**2.** 详细信息，请参见 **78K0/Kx2 Flash 存储器编程 (编程器) 应用注意 (U17739E)** 。

目标产品： **μPD78F0511(A), 78F0512(A), 78F0513(A), 78F0514(A), 78F0515(A)**

**最大额定值 ($T_A$ = 25°C) (1/2)**

| 参数 | 符号 | 条件 | | 额定值 | 单位 |
|---|---|---|---|---|---|
| 电源电压 | $V_{DD}$ | | | –0.5 ～ +6.5 | V |
| | $V_{SS}$ | | | –0.5 ～ +0.3 | V |
| | $AV_{REF}$ | | | –0.5 ～ $V_{DD}$ + 0.3[注1] | V |
| | $AV_{SS}$ | | | –0.5 ～ +0.3 | V |
| 输入电压 | $V_{I1}$ | P00, P01, P10 ～ P17, P20 ～ P27, P30 ～ P33, P40, P41, P70 ～ P73, P74[注2], P75[注2], P120 ～ P124, P140[注2], X1, X2, XT1, XT2, $\overline{RESET}$ | | –0.3 ～ $V_{DD}$ + 0.3[注1] | V |
| | $V_{I2}$ | P60 ～ P63 (N-ch 漏极开路) | | –0.3 ～ +6.5 | V |
| 输出电压 | $V_O$ | | | –0.3 ～ $V_{DD}$ + 0.3[注1] | V |
| 模拟输入电压 | $V_{AN}$ | ANI0 ～ ANI7 | | –0.3 ～ $AV_{REF}$ + 0.3[注1] 和 –0.3 ～ $V_{DD}$ + 0.3[注1] | V |
| 输出电流，高 | $I_{OH}$ | 每个引脚 | P00, P01, P10 ～ P17, P30 ～ P33, P40, P41, P70 ～ P73, P74[注2], P75[注2], P120, P130[注2], P140[注2] | –10 | mA |
| | | 所有引脚总和 –80 mA | P00, P01, P40, P41, P120, P130[注2], P140[注2] | –25 | mA |
| | | | P10 ～ P17, P30～ P33, P70 ～ P73, P74[注2], P75[注2] | –55 | mA |
| | | 每个引脚 | P20 ～ P27 | –0.5 | mA |
| | | 所有引脚总和 | | –2 | mA |
| | | 每个引脚 | P121 ～ P124 | –1 | mA |
| | | 所有引脚总和 | | –4 | mA |

注　**1.**　必须小于等于 6.5 V 。

　　**2.**　仅 48 引脚产品。

**注意事项**　**任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因而，必须保证产品在不超过最大额定值的条件下使用。**

**备注**　除非另外说明，复用功能引脚的特性与端口引脚的相同。

**(A) 级产品**

**最大额定值 ($T_A$ = 25°C) (2/2)**

| 参数 | 符号 | 条件 | | 额定值 | 单位 |
|---|---|---|---|---|---|
| 输出电流，低 | $I_{OL}$ | 每个引脚 | P00, P01, P10 ～ P17, P30 ～ P33, P40, P41, P60 ～ P63, P70～ P73, P74[注], P75[注], P120, P130[注], P140[注] | 30 | mA |
| | | 所有引脚总和 –200 mA | P00, P01, P40, P41, P120, P130[注], P140[注] | 60 | mA |
| | | | P10 ～ P17, P30～ P33, P60 ～ P63, P70～ P73, P74[注], P75[注] | 140 | mA |
| | | 每个引脚 | P20 ～ P27 | 1 | mA |
| | | 所有引脚总和 | | 5 | mA |
| | | 每个引脚 | P121 ～ P124 | 4 | mA |
| | | 所有引脚总和 | | 10 | mA |
| 工作环境温度 | $T_A$ | 正常操作模式 | | –40 ～ +85 | °C |
| | | Flash 存储器编程模式 | | | |
| 储藏温度 | $T_{stg}$ | | | –65 ～ +150 | °C |

注　　仅 48 引脚产品。

**注意事项**　**任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因而，必须保证产品在不超过最大额定值的条件下使用。**

备注　　除非另外说明，复用功能引脚的特性与端口引脚的相同。

**(A) 级产品**

**X1 振荡器特性**

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 振荡器 | 建议使用的电路 | 参数 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| 陶瓷振荡器 | $V_{SS}$ X1 X2 C1 C2 | X1 时钟振荡频率 ($f_X$) [注] | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 1.0 | | 20.0 | MHz |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 1.0 | | 10.0 | |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V | 1.0 | | 5.0 | |
| 晶体振荡器 | $V_{SS}$ X1 X2 C1 C2 | X1 时钟振荡频率 ($f_X$) [注] | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 1.0 | | 20.0 | MHz |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 1.0 | | 10.0 | |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V | 1.0 | | 5.0 | |

注　仅表示振荡器的特性。如需了解指令执行时间，可参见 **AC 特性** 。

注意事项 **1.** 在使用 **X1** 时钟时，上图中被虚线包围的部分的配线应按照如下配线方法配线，以防止连接线电容产生不利影响。

- 连接线越短越好。
- 连接线不应与其他信号线交叉。
- 流经的电流变化较大的信号线不要在振荡器周围布线。
- 要保持振荡器电容器的接地点电压与 $V_{SS}$ 相同。
- 不要将电容的地信号接入大电流地。
- 不要从振荡器获取信号。

**2.** 由于复位后 **CPU** 使用内部高速振荡时钟进行操作，因此用户要使用振荡稳定时间计数器的状态寄存器（**OSTC**）检测 **X1** 时钟振荡稳定时间。在充分评估了所使用的振荡器的振荡稳定时间后再确定 **OSTC** 和振荡稳定时间选择寄存器（**OSTS**）的振荡稳定时间。

**备注**　对于振荡器选择和振荡器常数，用户需要自己评估振荡器或要求制造商评估。

**(A) 级产品**

### 内部振荡器特性

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 振荡器 | 参数 | 条件 | | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| 8 MHz 内部振荡器 | 内部高速振荡时钟频率 ($f_{RH}$)[注] | RSTS = 1 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | 7.6 | 8.0 | 8.4 | MHz |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V | 7.6 | 8.0 | 10.4 | MHz |
| | | RSTS = 0 | | 2.48 | 5 | 9.86 | MHz |
| 240 kHz 内部振荡器 | 内部低速振荡时钟频率 ($f_{RL}$) | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | 216 | 240 | 264 | kHz |
| | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | 192 | 240 | 264 | kHz |

**注**　仅表示振荡器的特性。如需了解指令执行时间，可参见 **AC 特性** 。

**备注**　RSTS: 内部振荡模式寄存器 (RCM) 的第 7 位。

### XT1 荡器特性

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 振荡器 | 建议使用的电路 | 参数 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| 晶体振荡器 | $V_{SS}$ XT2 XT1 Rd C4 C3 | XT1 时钟振荡频率($f_X$)[注] | | 32 | 32.768 | 35 | kHz |

**注**　仅表示振荡器的特性。如需了解指令执行时间，可参见 **AC 特性** 。

**注意事项　1.** 在使用 **XT1** 时钟时，上图中被虚线包围的部分的配线应按照如下配线方法配线，以防止连接线电容产生不利影响。

- 连接线越短越好。
- 连接线不应与其他信号线交叉。
- 流经的电流变化较大的信号线不要在振荡器周围布线。
- 要保持振荡器电容器的接地点电压与 $V_{SS}$ 相同。
- 不要将电容的地信号接入大电流地。
- 不要从振荡器获取信号。

**2.** **XT1** 振荡器是一个低振幅电路，用于降低功耗，但由于比 **X1** 振荡器更容易受到噪音干扰，因此在使用 **XT1** 时钟时应特别注意连线方式。

**备注**　对于振荡器选择和振荡器常数，用户需要自己评估振荡器或要求制造商评估。

**(A) 级产品**

### DC 特性 (1/5)

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 参数 | 符号 | 条件 | | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| 输出电流，高[注1] | $I_{OH1}$ | P00, P01, P10 ～ P17, P30 ～ P33, P40, P41, P70 ～ P73, P74[注2], P75[注2], P120, P130[注2], P140[注2]的每个引脚 | 4.0V ≤ $V_{DD}$ ≤ 5.5V | | | –3.0 | mA |
| | | | 2.7V ≤ $V_{DD}$ < 4.0 V | | | –2.5 | mA |
| | | | 1.8V ≤ $V_{DD}$ < 2.7 V | | | –1.0 | mA |
| | | 引脚总和[注4] P00, P01, P40, P41, P120, P130[注2], P140[注2] | 4.0V ≤ $V_{DD}$ ≤ 5.5 V | | | –12.0 | mA |
| | | | 2.7V ≤ $V_{DD}$ < 4.0 V | | | –7.0 | mA |
| | | | 1.8V ≤ $V_{DD}$ < 2.7 V | | | –5.0 | mA |
| | | 引脚总和[注4] P10～ P17, P30 ～ P33, P70～ P73, P74[注2], P75[注2] | 4.0V ≤ $V_{DD}$ ≤ 5.5 V | | | –18.0 | mA |
| | | | 2.7V ≤ $V_{DD}$ < 4.0 V | | | –15.0 | mA |
| | | | 1.8V ≤ $V_{DD}$ < 2.7 V | | | –10.0 | mA |
| | | 所有引脚总和[注4] | 4.0V ≤ $V_{DD}$ ≤ 5.5 V | | | –23.0 | mA |
| | | | 2.7V ≤ $V_{DD}$ < 4.0 V | | | –20.0 | mA |
| | | | 1.8V ≤ $V_{DD}$ < 2.7 V | | | –15.0 | mA |
| | $I_{OH2}$ | P20 ～ P27 的每个引脚 | $AV_{REF}$ = $V_{DD}$ | | | –0.1 | mA |
| | | P121 ～ P124 的每个引脚 | | | | –0.1 | mA |
| 输出电流，低[注3] | $I_{OL1}$ | P00, P01, P10 ～ P17, P30 ～ P33, P40, P41, P70 ～ P73, P74[注2], P75[注2], P120, P130[注2], P140[注2]的每个引脚 | 4.0V ≤ $V_{DD}$ ≤ 5.5 V | | | 8.5 | mA |
| | | | 2.7V ≤ $V_{DD}$ < 4.0 V | | | 5.0 | mA |
| | | | 1.8V ≤ $V_{DD}$ < 2.7 V | | | 2.0 | mA |
| | | P60 ～ P63 的每个引脚 | 4.0V ≤ $V_{DD}$ ≤ 5.5 V | | | 15.0 | mA |
| | | | 2.7V ≤ $V_{DD}$ < 4.0 V | | | 5.0 | mA |
| | | | 1.8V ≤ $V_{DD}$ < 2.7 V | | | 2.0 | mA |
| | | 引脚总和[注4] P00, P01, P40, P41, P120, P130[注2], P140[注2] | 4.0V ≤ $V_{DD}$ ≤ 5.5 V | | | 20.0 | mA |
| | | | 2.7V ≤ $V_{DD}$ < 4.0 V | | | 15.0 | mA |
| | | | 1.8V ≤ $V_{DD}$ < 2.7 V | | | 9.0 | mA |
| | | 引脚总和[注4] P10 ～ P17, P30 ～ P33, P60 ～ P63, P70 ～ P73, P74[注2], P75[注2] | 4.0V ≤ $V_{DD}$ ≤ 5.5 V | | | 45.0 | mA |
| | | | 2.7V ≤ $V_{DD}$ < 4.0 V | | | 35.0 | mA |
| | | | 1.8V ≤ $V_{DD}$ < 2.7 V | | | 20.0 | mA |
| | | 所有引脚总和[注4] | 4.0V ≤ $V_{DD}$ ≤ 5.5 V | | | 65.0 | mA |
| | | | 2.7V ≤ $V_{DD}$ < 4.0 V | | | 50.0 | mA |
| | | | 1.8V ≤ $V_{DD}$ < 2.7 V | | | 29.0 | mA |

**注**

1. 即使电流从 $V_{DD}$ 流向输出引脚，设备操作时的电流值也可以得到保证。
2. 仅 48 引脚产品。
3. 即使电流从输出引脚流向 GND，设备操作时的电流值也可以得到保证。
4. 占空比=70%时 (输出电流的时间为 0.7 × t，不输出电流的时间为 0.3 × t, 其中 t 为指定时间)的规定。可通过以下表达式计算占空比不等于 70%时的引脚总的输出电流。
   - 当 $I_{OH}$ 的占空比为 n%: 引脚总的输出电流 = ($I_{OH}$ × 0.7)/(n × 0.01)

     <举例> 当占空比为 50%, $I_{OH}$ = 20.0 mA

     引脚总的输出电流 = (20.0 × 0.7)/(50 × 0.01) = 28.0 mA

   但是，允许流入每个引脚的电流不会随占空比而变化。高于最大额定值的电流不能流入引脚。

**备注** 除非另外说明，复用功能引脚的特性与端口引脚的相同。

**(A) 级产品**

**DC 特性 (2/5)**

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 参数 | 符号 | 条件 | | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| 输出电流，低[注 1] | $I_{OL2}$ | P20 ～ P27 的每个引脚 | $AV_{REF} = V_{DD}$ | | | 0.4 | mA |
| | | P121 ～ P124 的每个引脚 | | | | 0.4 | mA |
| 输入电压，高 | $V_{IH1}$ | P12, P13, P15, P40, P41, P121 ～ P124 | | $0.7V_{DD}$ | | $V_{DD}$ | V |
| | $V_{IH2}$ | P00, P01, P10, P11, P14, P16, P17, P30 ～ P33, P70 ～ P73, P74[注 2], P75[注 2], P120, P140[注 2], $\overline{RESET}$ | | $0.8V_{DD}$ | | $V_{DD}$ | V |
| | $V_{IH3}$ | P20 ～ P27 | $AV_{REF} = V_{DD}$ | $0.7AV_{REF}$ | | $AV_{REF}$ | V |
| | $V_{IH4}$ | P60 ～ P63 | | $0.7V_{DD}$ | | 6.0 | V |
| 输入电压，低 | $V_{IL1}$ | P12, P13, P15, P40, P41, P60 ～ P63, P121～P124 | | 0 | | $0.3V_{DD}$ | V |
| | $V_{IL2}$ | P00, P01, P10, P11, P14, P16, P17, P30 ～ P33, P70 ～ P73, P74[注 2], P75[注 2], P120, P140[注 2], $\overline{RESET}$ | | 0 | | $0.2V_{DD}$ | V |
| | $V_{IL3}$ | P20 ～ P27 | $AV_{REF} = V_{DD}$ | 0 | | $0.3AV_{REF}$ | V |
| 输出电压，高 | $V_{OH1}$ | P00, P01, P10 ～ P17, P30 ～ P33, P40, P41, P70 ～ P73, P74[注 2], P75[注 2], P120, P130[注 2], P140[注 2] | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $I_{OH1}$ = –3.0 mA | $V_{DD}$ – 0.7 | | | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $I_{OH1}$ = –2.5 mA | $V_{DD}$ – 0.5 | | | V |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V, $I_{OH1}$ = –1.0 mA | $V_{DD}$ – 0.5 | | | V |
| | $V_{OH2}$ | P20 ～ P27 | $AV_{REF} = V_{DD}$, $I_{OH2}$ = –100 μA | $V_{DD}$ – 0.5 | | | V |
| | | P121 ～ P124 | $I_{OH2}$ = –100 μA | $V_{DD}$ – 0.5 | | | V |

注　**1.** 即使电流从输出引脚流向 GND，设备操作时的电流值也可以得到保证。
　　**2.** 仅 48 引脚产品。

备注　除非另外说明，复用功能引脚的特性与端口引脚的相同。

**(A) 级产品**

## DC 特性 (3/5)

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 参数 | 符号 | 条件 | | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| 输出电压，低 | $V_{OL1}$ | P00, P01, P10 ～ P17, P30 ～ P33, P40, P41, P70 ～ P73, P74[注], P75[注], P120, P130[注], P140[注] | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $I_{OL1}$ = 8.5 mA | | | 0.7 | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $I_{OL1}$ = 5.0 mA | | | 0.7 | V |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V, $I_{OH1}$ = 2.0 mA | | | 0.5 | V |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V, $I_{OL1}$ = 0.5 mA | | | 0.4 | V |
| | $V_{OL2}$ | P20 ～ P27 | $AV_{REF}$ = $V_{DD}$, $I_{OL2}$ = 0.4 mA | | | 0.4 | V |
| | | P121 ～ P124 | $I_{OL2}$ = 0.4 mA | | | 0.4 | V |
| | $V_{OL3}$ | P60 ～ P63 | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $I_{OL1}$ = 15.0 mA | | | 2.0 | V |
| | | | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $I_{OL1}$ = 5.0 mA | | | 0.4 | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $I_{OL1}$ = 5.0 mA | | | 0.6 | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $I_{OL1}$ = 3.0 mA | | | 0.4 | V |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V, $I_{OL1}$ = 2.0 mA | | | 0.4 | V |
| 输入泄漏电流，高 | $I_{LIH1}$ | P00, P01, P10 ～ P17, P30 ～ P33, P40, P41, P60 ～ P63, P70 ～ P73, P74[注], P75[注], P120, P140[注] | $V_I$ = $V_{DD}$ | | | 1 | μA |
| | $I_{LIH2}$ | P20 ～ P27 | $V_I$ = $AV_{REF}$ = $V_{DD}$ | | | 1 | μA |
| | $I_{LIH3}$ | P121 ～ 124 (X1, X2, XT1, XT2) | $V_I$ = $V_{DD}$ I/O 端口模式 | | | 1 | μA |
| | | | $V_I$ = $V_{DD}$ OSC 模式 | | | 20 | μA |
| 输入泄漏电流，低 | $I_{LIL1}$ | P00, P01, P10 ～ P17, P30 ～ P33, P40, P41, P60 ～ P63, P70 ～ P73, P74[注], P75[注], P120, P140[注] | $V_I$ = $V_{SS}$ | | | –1 | μA |
| | $I_{LIL2}$ | P20 ～ P27 | $V_I$ = $V_{SS}$, $AV_{REF}$ = $V_{DD}$ | | | –1 | μA |
| | $I_{LIL3}$ | P121 ～ 124 (X1, X2, XT1, XT2) | $V_I$ = $V_{SS}$ I/O 端口模式 | | | –1 | μA |
| | | | $V_I$ = $V_{SS}$ OSC 模式 | | | –20 | μA |
| 上拉电阻 | $R_U$ | $V_I$ = $V_{DD}$ | | 10 | 20 | 100 | kΩ |
| FLMD0 电压 | $V_{IL}$ | 正常操作模式 | | 0 | | 0.2$V_{DD}$ | V |
| | $V_{IH}$ | 自编程模式 | | 0.8$V_{DD}$ | | $V_{DD}$ | V |
| 外部时钟输入电压 | $V_{IL}$ | P122, P124 | 外部时钟输入模式 | 0 | | 0.2$V_{DD}$ | V |
| | $V_{IH}$ | | | 0.8$V_{DD}$ | | $V_{DD}$ | V |

**注**　仅 48 引脚产品。

**备注**　除非另外说明，复用功能引脚的特性与端口引脚的相同。

**(A) 级产品**

**DC 特性 (4/5)**

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 参数 | 符号 | 条件 | | | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|---|
| 电源电流 | $I_{DD1}$ [注 1] | 运行模式 | $f_{XH}$ = 20 MHz[注 2], $V_{DD}$ = 5.0 V | 方波输入 | | 3.2 | 5.5 | mA |
| | | | | 振荡器连接 | | 4.5 | 6.9 | |
| | | | $f_{XH}$ = 10 MHz[注 2, 3], $V_{DD}$ = 5.0 V | 方波输入 | | 1.6 | 2.8 | mA |
| | | | | 振荡器连接 | | 2.3 | 3.9 | |
| | | | $f_{XH}$ = 10 MHz[注 2, 3], $V_{DD}$ = 3.0 V | 方波输入 | | 1.5 | 2.7 | mA |
| | | | | 振荡器连接 | | 2.2 | 3.2 | |
| | | | $f_{XH}$ = 5 MHz[注 2, 3], $V_{DD}$ = 3.0 V | 方波输入 | | 0.9 | 1.6 | mA |
| | | | | 振荡器连接 | | 1.3 | 2.0 | |
| | | | $f_{XH}$ = 5 MHz[注 2, 3], $V_{DD}$ = 2.0 V | 方波输入 | | 0.7 | 1.4 | mA |
| | | | | 振荡器连接 | | 1.0 | 1.6 | |
| | | | $f_{RH}$ = 8 MHz, $V_{DD}$ = 5.0 V | | | 1.4 | 2.5 | mA |
| | | | $f_{SUB}$ = 32.768 kHz[注 2, 4], $V_{DD}$ = 5.0 V | 方波输入 | | 6 | 25 | μA |
| | | | | 振荡器连接 | | 15 | 30 | |
| | $I_{DD2}$ [注 5] | HALT 模式 | $f_{XH}$ = 20 MHz[注 2], $V_{DD}$ = 5.0 V | 方波输入 | | 0.8 | 2.6 | mA |
| | | | | 振荡器连接 | | 2.0 | 4.4 | |
| | | | $f_{XH}$ = 10 MHz[注 2, 3], $V_{DD}$ = 5.0 V | 方波输入 | | 0.4 | 1.3 | mA |
| | | | | 振荡器连接 | | 1.0 | 2.4 | |
| | | | $f_{XH}$ = 5 MHz[注 2, 3], $V_{DD}$ = 3.0 V | 方波输入 | | 0.2 | 0.65 | mA |
| | | | | 振荡器连接 | | 0.5 | 1.1 | |
| | | | $f_{RH}$ = 8 MHz, $V_{DD}$ = 5.0 V | | | 0.4 | 1.2 | mA |
| | | | $f_{SUB}$ = 32.768 kHz[注 2, 4], $V_{DD}$ = 5.0 V | 方波输入 | | 3.0 | 22 | μA |
| | | | | 振荡器连接 | | 12 | 25 | |
| | $I_{DD3}$ [注 5] | STOP 模式 | $V_{DD}$ = 5.0 V | | | 1 | 20 | μA |
| | | | $V_{DD}$ = 5.0 V, $T_A$ = –40 ～ +70°C | | | 1 | 10 | μA |

注
1. 所有内部电源($V_{DD}$, $AV_{REF}$)电流，包括外围操作电流（而不包括电流流入端口的上拉电阻，A/D 转换器的部分）。
2. 不包括内部高速时钟($f_{RH}$)的操作电流。
3. 当 AMPH (时钟操作模式选择寄存器 (OSCCTL) 的第 0 位 ) = 0 。
4. 当停止主系统时钟时。
5. 所有内部电源($V_{DD}$, $AV_{REF}$)电流，包括外围操作电流（而不包括电流流入端口的上拉电阻，A/D 转换器，看门狗定时器，和 LVI 电路的部分）。

备注
1. $f_{XH}$: 高速系统时钟频率 (X1 时钟振荡频率或外部主系统时钟频率)
2. $f_{RH}$: 内部高速振荡时钟频率
3. $f_{SUB}$: 副系统时钟频率 (XT1 时钟振荡频率或外部副系统时钟频率)

**(A) 级产品**

**DC 特性 (5/5)**

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 参数 | 符号 | 条件 | | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| A/D 转换器工作电流 | $I_{ADC}$ 注 1 | 在最大转换速度时 | 2.3 V ≤ $AV_{REF}$ ≤ $V_{DD}$ | | 0.86 | 1.9 | mA |
| | | 非转换时 (比较器操作) 注 2 | | | 0.6 | 1.2 | mA |
| 看门狗定时器工作电流 | $I_{WDT}$ 注 3 | 在 240 kHz 内部低速振荡时钟下工作 | | | 5 | 10 | μA |
| LVI 工作电流 | $I_{LVI}$ 注 4 | | | | 9 | 18 | μA |

**注**

1. 电流仅流入 A/D 转换器中。在操作或 HALT 模式下当 A/D 转换器工作时 78K0/KC2 电流值为 $I_{DD1}$ 或 $I_{DD2}$ 和 $I_{ADC}$ 之和。
2. 当 ADCE (A/D 转换器模式寄存器 (ADM) 的第 0 位 ) = 1 并且 ADCS (ADM 的第 7 位) = 0 时的电流。
3. 电流流入看门狗定时器中。在 HALT 或 STOP 模式下当看门狗定时器工作时 78K0/KC2 电流值为 $I_{DD2}$ 或 $I_{DD3}$ 和 $I_{WDT}$ 之和。
4. 电流流入 LVI 电路。在 HALT 或 STOP 模式下当 LVI 电路工作时 78K0/KC2 电流值为 $I_{DD2}$ 或 $I_{DD3}$ 和 $I_{LVI}$ 之和。

(A) 级产品

## AC 特性

### (1) 基本操作

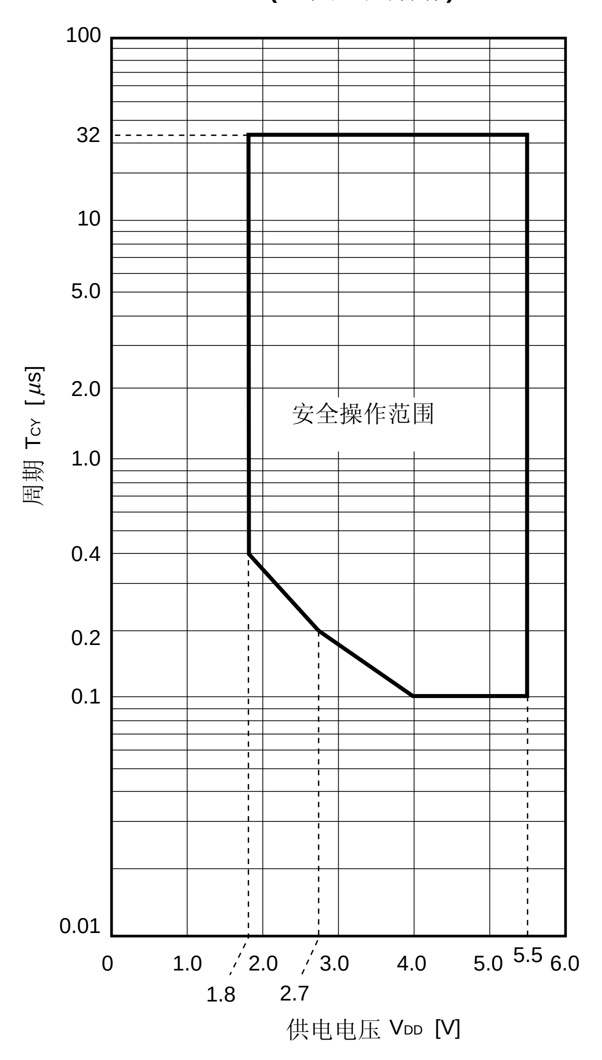
**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

| 参数 | 符号 | 条件 | | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| 指令周期（最小指令执行时间） | $T_{CY}$ | 主系统时钟 ($f_{XP}$) 操作 | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0.1 | | 32 | μs |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0.2 | | 32 | μs |
| | | | 1.8 V ≤ $V_{DD}$ < 2.7 V | 0.4 注 1 | | 32 | μs |
| | | 副系统时钟 ($f_{SUB}$) 操作 | | 114 | 122 | 125 | μs |
| 外部主系统时钟频率 | $f_{EXCLK}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.0 | | 20.0 | MHz |
| | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | 1.0 | | 10.0 | MHz |
| | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | 1.0 | | 5.0 | MHz |
| 外部主系统时钟输入高电平宽度，低电平宽度 | $t_{EXCLKH}$, $t_{EXCLKL}$ | | | $(1/f_{EXCLK} \times 1/2) - 1$ | | | ns |
| 外部副系统时钟频率 | $f_{EXCLKS}$ | | | 32 | 32.768 | 35 | kHz |
| 外部副系统时钟输入高电平宽度，低电平宽度 | $t_{EXCLKSH}$, $t_{EXCLKSL}$ | | | $(1/f_{EXCLKS} \times 1/2) - 5$ | | | ns |
| TI000, TI010 输入高电平宽度，低电平宽度 | $t_{TIH0}$, $t_{TIL0}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | $2/f_{sam}$ + 0.1 注 2 | | | μs |
| | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | $2/f_{sam}$ + 0.2 注 2 | | | μs |
| | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | $2/f_{sam}$ + 0.5 注 2 | | | μs |
| TI50, TI51 输入频率 | $f_{TI5}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | | | 10 | MHz |
| | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | | | 10 | MHz |
| | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | | | 5 | MHz |
| TI50, TI51 输入高电平宽度，低电平宽度 | $t_{TIH5}$, $t_{TIL5}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | 50 | | | ns |
| | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | 50 | | | ns |
| | | 1.8 V ≤ $V_{DD}$ < 2.7 V | | 100 | | | ns |
| 中断输入高电平宽度，低电平宽度 | $t_{INTH}$, $t_{INTL}$ | | | 1 | | | μs |
| 按键中断输入低电平宽度 | $t_{KR}$ | | | 250 | | | ns |
| $\overline{RESET}$ 低电平宽度 | $t_{RSL}$ | | | 10 | | | μs |

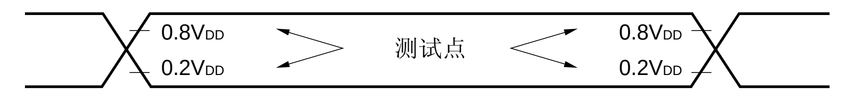
注　**1.** 当使用 8 MHz 内部振荡器时为 0.38 μs。

**2.** 使用预分频模式寄存器 00（PRM00）的第 0 和 1 位（PRM000、PRM001）可以选择 $f_{sam} = f_{PRS}$, $f_{PRS}/4$, $f_{PRS}/256$。注意，当选择 TI000 的有效沿作为计数时钟时，$f_{sam} = f_{PRS}$ 。

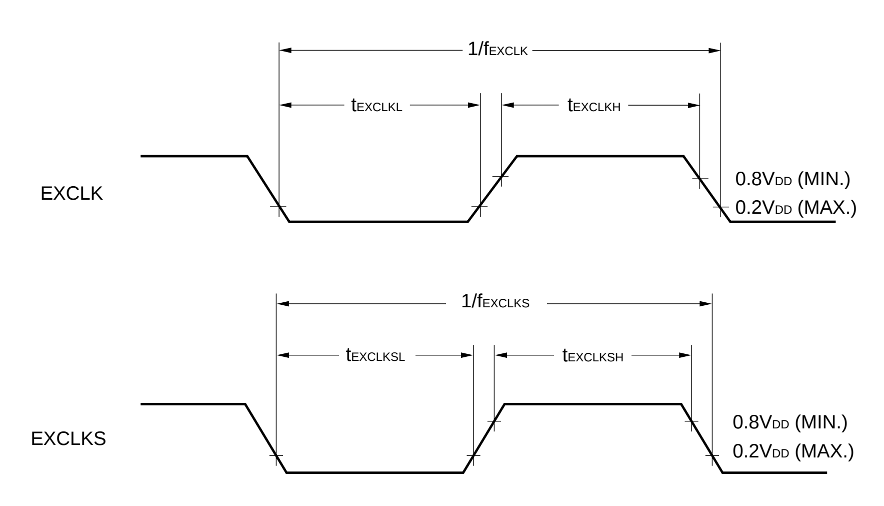
**$T_{CY}$ vs. $V_{DD}$ (主系统时钟操作)**



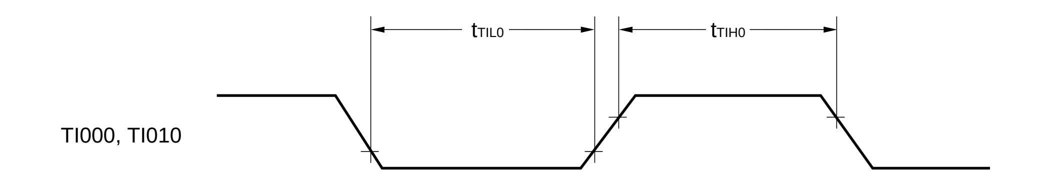

**AC 时序测试点 (外部主系统时钟和外部副系统时钟除外)**

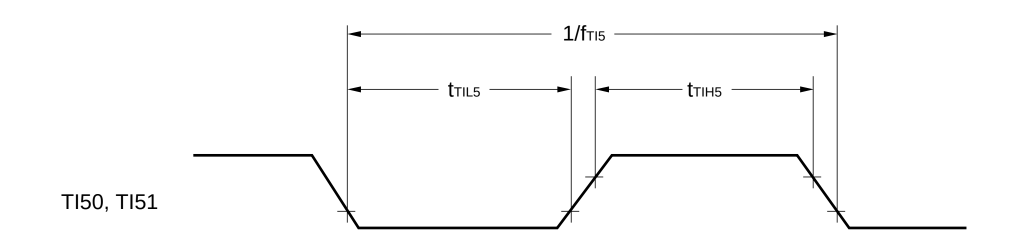



外部主系统时钟时序、外部副系统时钟时序

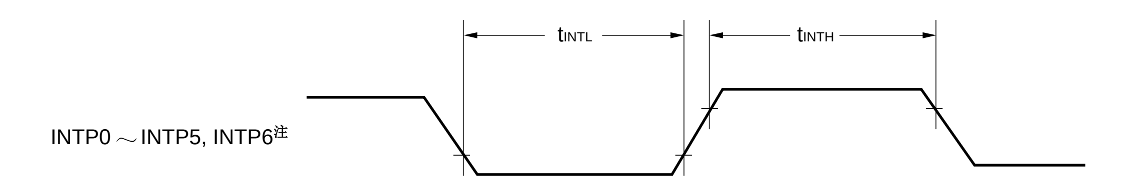


**TI 时序**



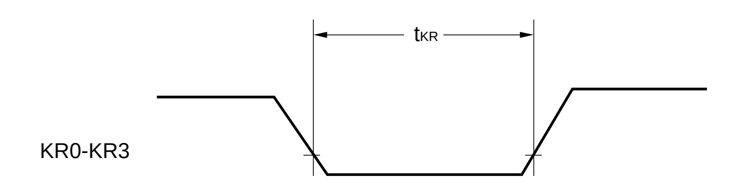




中断请求输入时序

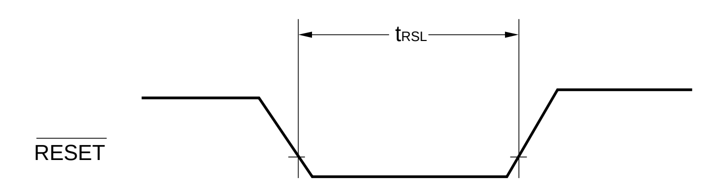



注 仅 48 引脚产品。

按键中断输入时序




**$\overline{\text{RESET}}$** 输入时序



**(A) 级产品**

**(2) 串行接口**

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

**(a) UART6 (专用波特率发生器输出)**

| 参数 | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|
| 传输速率 | | | | | 625 | kbps |

**(b) UART0 (专用波特率发生器输出)**

| 参数 | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|
| 传输速率 | | | | | 625 | kbps |

**(c) IIC0**

| 参数 | 符号 | 标准模式 | | 高速模式 | | 单位 |
|---|---|---|---|---|---|---|
| | | MIN. | MAX. | MIN. | MAX. | |
| SCL0 时钟频率 | $f_{SCL}$ | 0 | 100 | 0 | 400 | kHz |
| 启动/重新启动条件的建立时间[注 1] | $t_{SU:STA}$ | 4.8 | – | 0.7 | – | μs |
| 保持时间 | $t_{HD:STA}$ | 4.1 | – | 0.7 | – | μs |
| 当 SCL0 = “L”时的保持时间 | $t_{LOW}$ | 5.0 | – | 1.25 | – | μs |
| 当 SCL0 = “H”时的保持时间 | $t_{HIGH}$ | 5.0 | – | 1.25 | – | μs |
| 数据建立时间 (接收) | $t_{SU:DAT}$ | 0 | – | 0 | – | μs |
| 数据保持时间 (发送)[注 2] | $t_{HD:DAT}$ | 0.47 | 4.0 | 0.23 | 1.00 | μs |

注 **1.** 当检测到启动/重新启动条件时在经历了该周期时间后产生第一个时钟脉冲。

**2.** $t_{HD:DAT}$ 的最大值 (MAX.) 是在正常传送期间，在 $\overline{ACK}$ (应答)时序中插入一个等待状态。

### (d) CSI10 (主设备模式, $\overline{\text{SCK10}}$... 内部时钟输出)

| 参数 | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|
| $\overline{\text{SCK10}}$ 周期时间 | $t_{KCY1}$ | $4.0\ V \le V_{DD} \le 5.5\ V$ | 100 | | | ns |
| | | $2.7\ V \le V_{DD} < 4.0\ V$ | 200 | | | ns |
| | | $1.8\ V \le V_{DD} < 2.7\ V$ | 400 | | | ns |
| $\overline{\text{SCK10}}$ 高/低电平宽度 | $t_{KH1}$,<br>$t_{KL1}$ | | $t_{KCY1}/2 - 10$[注1] | | | ns |
| SI10 建立时间 (到 $\overline{\text{SCK10}}$↑) | $t_{SIK1}$ | | 30 | | | ns |
| SI10 保持时间 (从 $\overline{\text{SCK10}}$↑) | $t_{KSI1}$ | | 30 | | | ns |
| 从 $\overline{\text{SCK10}}$↓到 SO10 输出的延迟时间 | $t_{KSO1}$ | C = 50 pF[注2] | | | 40 | ns |

注 **1.** 使用高速系统时钟 ($f_{XH}$) 时的取值

**2.** C 是 $\overline{\text{SCK10}}$ 和 SO10 输出线的负载电容。

### (e) CSI10 (从设备模式, $\overline{\text{SCK1n}}$... 外部时钟输入)

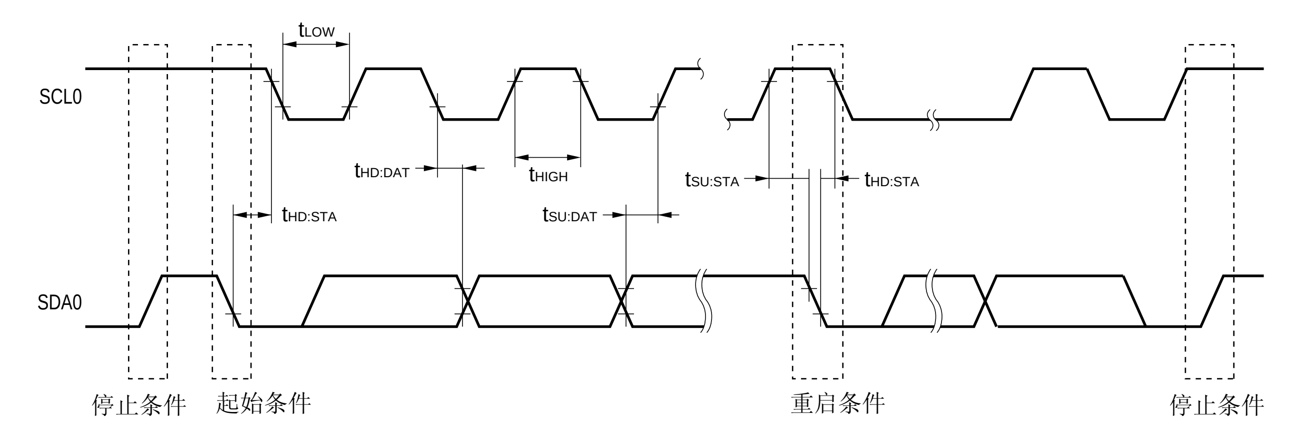
| 参数 | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|
| $\overline{\text{SCK10}}$ 周期 | $t_{KCY2}$ | | 400 | | | ns |
| $\overline{\text{SCK10}}$ 高/低电平宽度 | $t_{KH2}$,<br>$t_{KL2}$ | | $t_{KCY2}/2$ | | | ns |
| SI10 建立时间 (至 $\overline{\text{SCK10}}$↑) | $t_{SIK2}$ | | 80 | | | ns |
| SI10 保持时间 (从 $\overline{\text{SCK10}}$↑) | $t_{KSI2}$ | | 50 | | | ns |
| 从 $\overline{\text{SCK10}}$↓ 到 SO10 输出的延迟时间 | $t_{KSO2}$ | C = 50 pF[注] | | | 120 | ns |

注 C 是 SO10 输出线的负载电容。

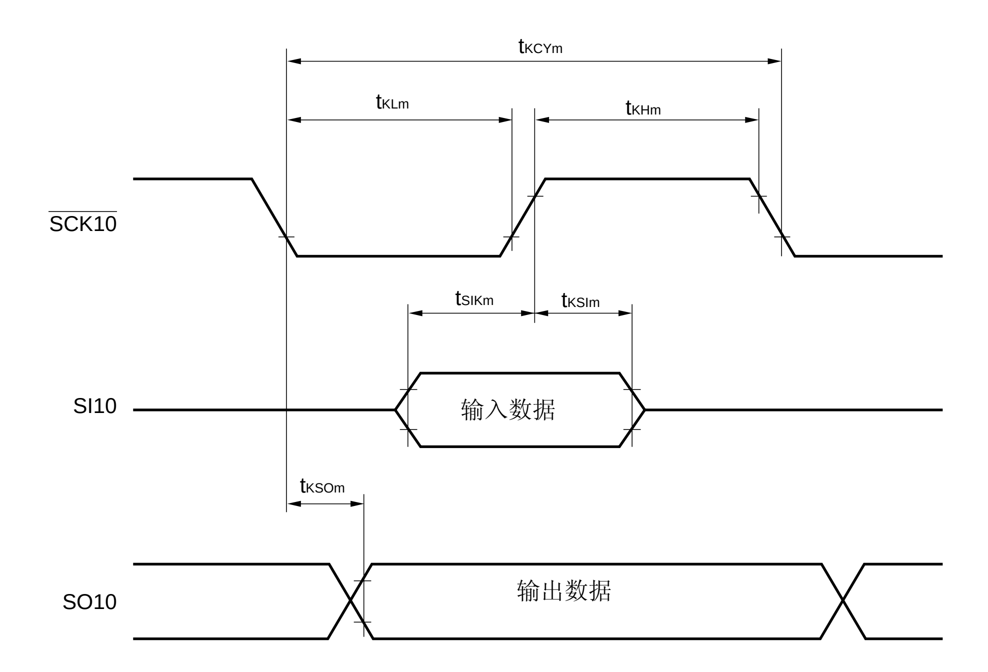
(A) 级产品

串行传输时序

**IIC0:**




**CSI10:**




备注　　m = 1, 2

**A/D 转换器特性**

**($T_A$ = –40 ～ +85°C, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V, 2.3 V ≤ $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

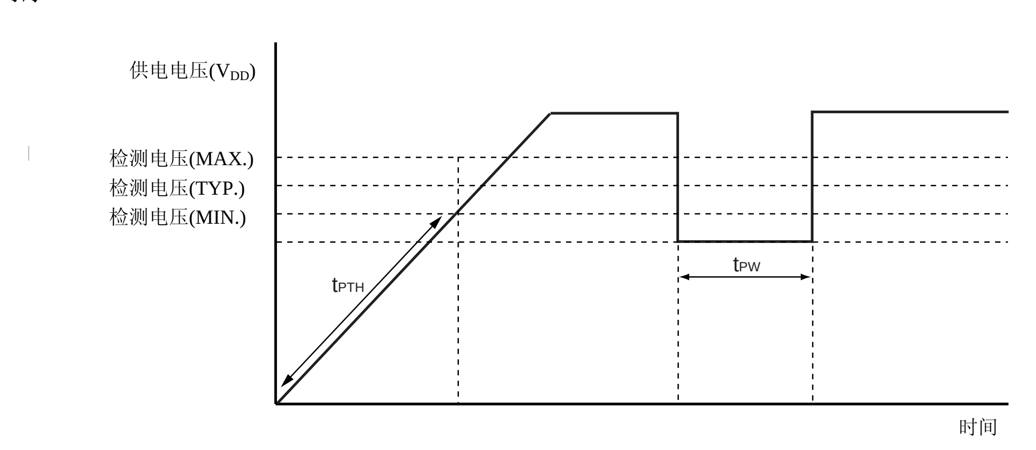
| 参数 | 符号 | 条件 | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| 分辨率 | $R_{ES}$ | | | | 10 | bit |
| 总误差注 [1, 2] | $A_{INL}$ | 4.0 V ≤ $AV_{REF}$ ≤ 5.5 V | | | ±0.4 | %FSR |
| | | 2.7 V ≤ $AV_{REF}$ < 4.0 V | | | ±0.6 | %FSR |
| | | 2.3 V ≤ $AV_{REF}$ < 2.7 V | | | ±1.2 | %FSR |
| 转换时间 | $t_{CONV}$ | 4.0 V ≤ $AV_{REF}$ ≤ 5.5 V | 6.1 | | 36.7 | μs |
| | | 2.7 V ≤ $AV_{REF}$ < 4.0 V | 12.2 | | 36.7 | μs |
| | | 2.3 V ≤ $AV_{REF}$ < 2.7 V | 27 | | 66.6 | μs |
| 零度误差注 [1, 2] | $E_{ZS}$ | 4.0 V ≤ $AV_{REF}$ ≤ 5.5 V | | | ±0.4 | %FSR |
| | | 2.7 V ≤ $AV_{REF}$ < 4.0 V | | | ±0.6 | %FSR |
| | | 2.3 V ≤ $AV_{REF}$ < 2.7 V | | | ±0.6 | %FSR |
| 满度误差注 [1, 2] | $E_{FS}$ | 4.0 V ≤ $AV_{REF}$ ≤ 5.5 V | | | ±0.4 | %FSR |
| | | 2.7 V ≤ $AV_{REF}$ < 4.0 V | | | ±0.6 | %FSR |
| | | 2.3 V ≤ $AV_{REF}$ < 2.7 V | | | ±0.6 | %FSR |
| 积分非线性误差注 [1] | $I_{LE}$ | 4.0 V ≤ $AV_{REF}$ ≤ 5.5 V | | | ±2.5 | LSB |
| | | 2.7 V ≤ $AV_{REF}$ < 4.0 V | | | ±4.5 | LSB |
| | | 2.3 V ≤ $AV_{REF}$ < 2.7 V | | | ±6.5 | LSB |
| 微分非线性误差注 [1] | $D_{LE}$ | 4.0 V ≤ $AV_{REF}$ ≤ 5.5 V | | | ±1.5 | LSB |
| | | 2.7 V ≤ $AV_{REF}$ < 4.0 V | | | ±2.0 | LSB |
| | | 2.3 V ≤ $AV_{REF}$ < 2.7 V | | | ±2.0 | LSB |
| 模拟输入电压 | $V_{AIN}$ | | $AV_{SS}$ | | $AV_{REF}$ | V |

注　**1.** 不包括量化误差 (±1/2 LSB)
　　**2.** 该值表示满度的百分比 (%FSR)

**1.59 V POC 电路特性 ($T_A$ = –40 ～ +85°C, $V_{SS}$ = 0 V)**

| 参数 | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|
| 检测电压 | $V_{POC}$ | | 1.44 | 1.59 | 1.74 | V |
| 供电电压的上升斜率 | $t_{PTH}$ | $V_{DD}$: 0 V →变到 $V_{POC}$ 的指定电压 | 0.5 | | | V/ms |
| 最小脉冲宽度 | $t_{PW}$ | | 200 | | | μs |

**POC 电路时序**



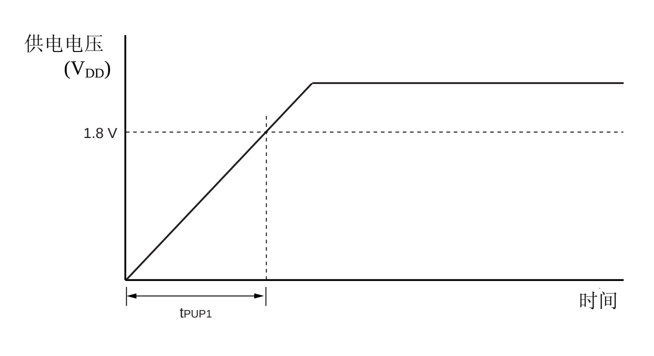
**(A) 级产品**

供电电压上升时间 ($T_A$ = –40 ～ +85°C, $V_{SS}$ = 0 V)

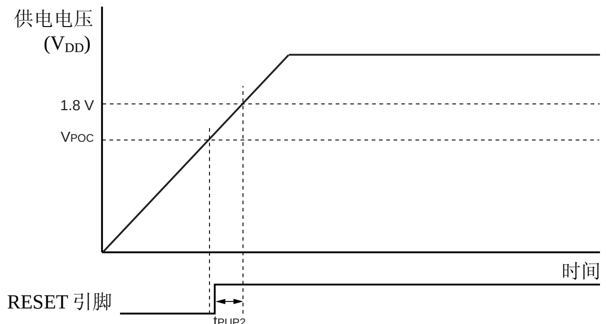
| 参数 | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|
| 上升至 1.8 V ($V_{DD}$ (MIN.))所需的最长时间 ($V_{DD}$: 0 V → 1.8 V) | $t_{PUP1}$ | 当没有使用 $\overline{RESET}$ 输入时，POCMODE (选项字节) = 0 | | | 3.6 | ms |
| 上升至 1.8 V ($V_{DD}$ (MIN.))所需的最长时间 (释放 $\overline{RESET}$ 输入 → $V_{DD}$: 1.8 V) | $t_{PUP2}$ | 当使用 $\overline{RESET}$ 输入时，POCMODE (选项字节) = 0 | | | 1.9 | ms |

供电电压上升时间时序

• 当没有使用 $\overline{RESET}$ 引脚输入时




•当使用 $\overline{RESET}$ 引脚输入时




**2.7 V POC 电路特性($T_A$ = –40 ～ +85°C, $V_{SS}$ = $EV_{SS}$ = 0 V)**

| 参数 | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|
| 供电电压应用中的检测电压 | $V_{DDPOC}$ | POCMODE (选项字节) = 1 | 2.50 | 2.70 | 2.90 | V |

**(A) 级产品**

**LVI 电路特性 ($T_A$ = –40 ～ +85°C, $V_{POC} \le V_{DD} \le$ 5.5 V, $AV_{REF} \le V_{DD}$, $V_{SS}$ = 0 V)**

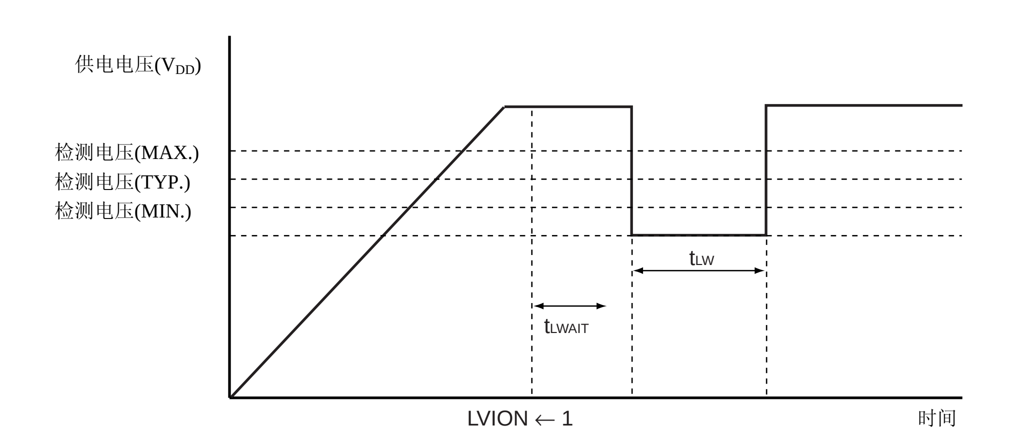
| 参数 | | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| 检测电压 | 供电电压电平 | $V_{LVI0}$ | | 4.14 | 4.24 | 4.34 | V |
| | | $V_{LVI1}$ | | 3.99 | 4.09 | 4.19 | V |
| | | $V_{LVI2}$ | | 3.83 | 3.93 | 4.03 | V |
| | | $V_{LVI3}$ | | 3.68 | 3.78 | 3.88 | V |
| | | $V_{LVI4}$ | | 3.52 | 3.62 | 3.72 | V |
| | | $V_{LVI5}$ | | 3.37 | 3.47 | 3.57 | V |
| | | $V_{LVI6}$ | | 3.22 | 3.32 | 3.42 | V |
| | | $V_{LVI7}$ | | 3.06 | 3.16 | 3.26 | V |
| | | $V_{LVI8}$ | | 2.91 | 3.01 | 3.11 | V |
| | | $V_{LVI9}$ | | 2.75 | 2.85 | 2.95 | V |
| | | $V_{LVI10}$ | | 2.60 | 2.70 | 2.80 | V |
| | | $V_{LVI11}$ | | 2.45 | 2.55 | 2.65 | V |
| | | $V_{LVI12}$ | | 2.29 | 2.39 | 2.49 | V |
| | | $V_{LVI13}$ | | 2.14 | 2.24 | 2.34 | V |
| | | $V_{LVI14}$ | | 1.98 | 2.08 | 2.18 | V |
| | | $V_{LVI15}$ | | 1.83 | 1.93 | 2.03 | V |
| | 外部输入引脚[注 1] | EXLVI | EXLVI < $V_{DD}$, 1.8 V ≤ $V_{DD}$ ≤ 5.5 V | 1.11 | 1.21 | 1.31 | V |
| 最短脉冲宽度 | | | | 200 | | | μs |
| 操作稳定等待时间[注 2] | | | | | | 10 | μs |

注 **1.** 使用 EXLVI/P120/INTP0 引脚

**2.** 从设置低电压检测寄存器(LVIM)的第 7 位(LVION)=1 到操作稳定所需的时间。

备注 $V_{LVI(n-1)} > V_{LVIn}$: n = 1 ～ 15

**LVI 电路时序**

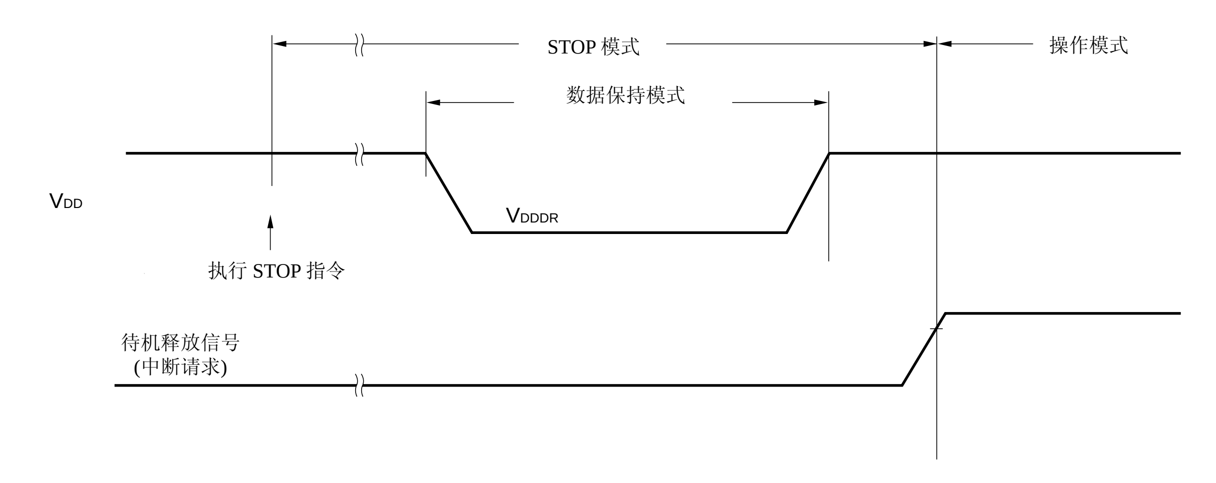


**(A) 级产品**

**数据存储器在 STOP 模式下低供电电压时的数据保持特性 ($T_A$ = –40 ～ +85°C)**

| 参数 | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|
| 数据保持供电电压 | $V_{DDDR}$ | | 1.44[注] | | 5.5 | V |

注 该值根据 POC 检测电压而变化。当电压降低时，一直保持该数据直至 POC 复位有效，但当 POC 复位有效时数据不能保持。




**Flash 存储器编程特性**
**($T_A$ = –40 ～ +85°C, 2.7 V ≤ $V_{DD}$ ≤ 5.5 V, $AV_{REF}$ ≤ $V_{DD}$, $V_{SS}$ = $AV_{SS}$ = 0 V)**

• **基本特性**

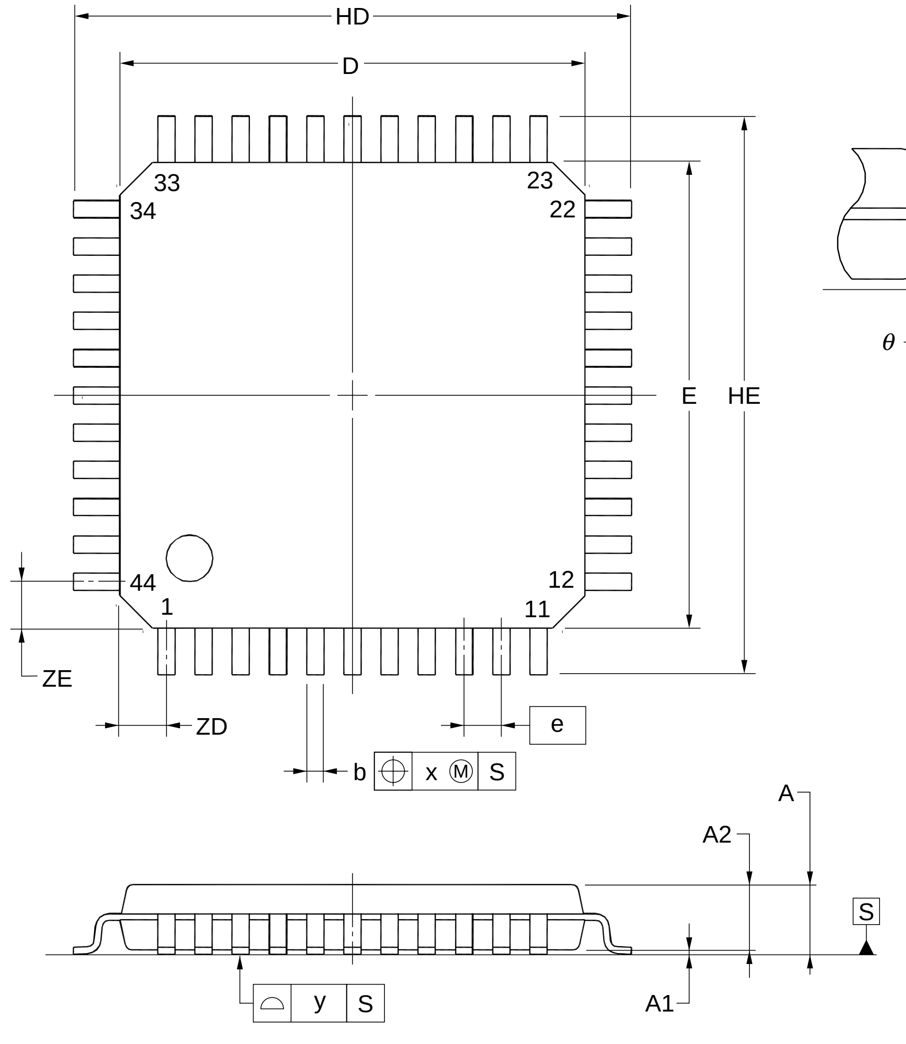
| 参数 | | 符号 | 条件 | MIN. | TYP. | MAX. | 单位 |
|---|---|---|---|---|---|---|---|
| $V_{DD}$ 供电电流 | | $I_{DD}$ | $f_{XP}$ = 10 MHz (TYP.), 20 MHz (MAX.) | | 4.5 | 11.0 | mA |
| 擦除时间[注 1, 2] | 所有 block | $T_{eraca}$ | | | 20 | 200 | ms |
| | Block 单元 | $T_{erasa}$ | | | 20 | 200 | ms |
| 写时间(以 8 位为单位)[注 1] | | $T_{wrwa}$ | | | 10 | 100 | μs |
| 每片重写次数 | | $C_{erwr}$ | 保持时间: 10 年<br>擦除 1 次 +擦除后写 1 次= 1 次重写[注 3] | T.B.D. | | | 次数 |

注 **1.** flash 存储器的特性。当使用专用 flash 存储器编程器，PG-FP4 时，会应用这些特性，并且自编程下的重写次数，参看**表 25-12** 和 **25-13**。
**2.** 不包括擦除前的预写时间和擦除验证时间(写回时间)。
**3.** 在出货后第一次对产品进行写操作时，“擦除 → 写”和 “只写”作为一次重写。

备注 **1.** $f_{XP}$: 主系统时钟振荡频率
**2.** 详细信息，请参见 **78K0/Kx2 Flash 存储器编程 (编程器) 应用注意 (U17739E)**。

- **μPD78F0511GB-UES-A, 78F0512GB-UES-A, 78F0513GB-UES-A, 78F0513DGB-UES-A**

## 44 引脚塑封 LQFP(10×10)




(UNIT:mm)

| 项目 | 尺寸 |
|---|---|
| D | 10.00±0.20 |
| E | 10.00±0.20 |
| HD | 12.00±0.20 |
| HE | 12.00±0.20 |
| A | 1.60 MAX. |
| A1 | 0.10±0.05 |
| A2 | 1.40±0.05 |
| A3 | 0.25 |
| b | $0.37^{+0.08}_{-0.07}$ |
| c | $0.145^{+0.055}_{-0.045}$ |
| L | 0.50 |
| Lp | 0.60±0.15 |
| L1 | 1.00±0.20 |
| $\theta$ | $3°^{+5°}_{-3°}$ |
| e | 0.80 |
| x | 0.20 |
| y | 0.10 |
| ZD | 1.00 |
| ZE | 1.00 |

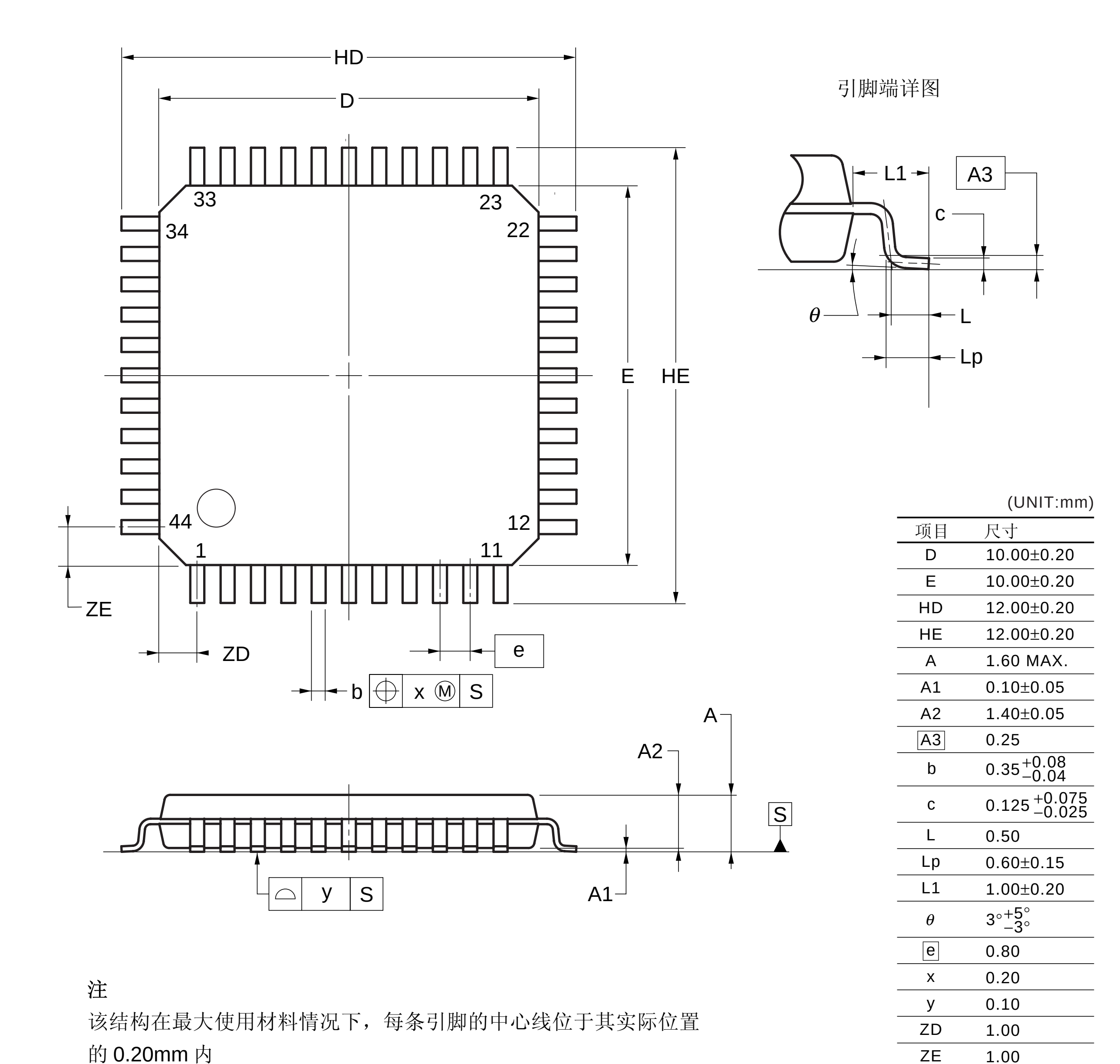
**P44GB-80-UES-1**

注

该结构在最大使用用材料情况下，每条引脚的中心线位于其实际位置的 0.20mm 内

<R> ● **μPD78F0511GB(A)-GAF-AX, 78F0512GB(A)-GAF-AX, 78F0513GB(A)-GAF-AX**

## 44 引脚塑封 LQFP(10×10)




(UNIT:mm)

| 项目 | 尺寸 |
|---|---|
| D | 10.00±0.20 |
| E | 10.00±0.20 |
| HD | 12.00±0.20 |
| HE | 12.00±0.20 |
| A | 1.60 MAX. |
| A1 | 0.10±0.05 |
| A2 | 1.40±0.05 |
| A3 | 0.25 |
| b | $0.35^{+0.08}_{-0.04}$ |
| c | $0.125^{+0.075}_{-0.025}$ |
| L | 0.50 |
| Lp | 0.60±0.15 |
| L1 | 1.00±0.20 |
| θ | $3°^{+5°}_{-3°}$ |
| e | 0.80 |
| x | 0.20 |
| y | 0.10 |
| ZD | 1.00 |
| ZE | 1.00 |

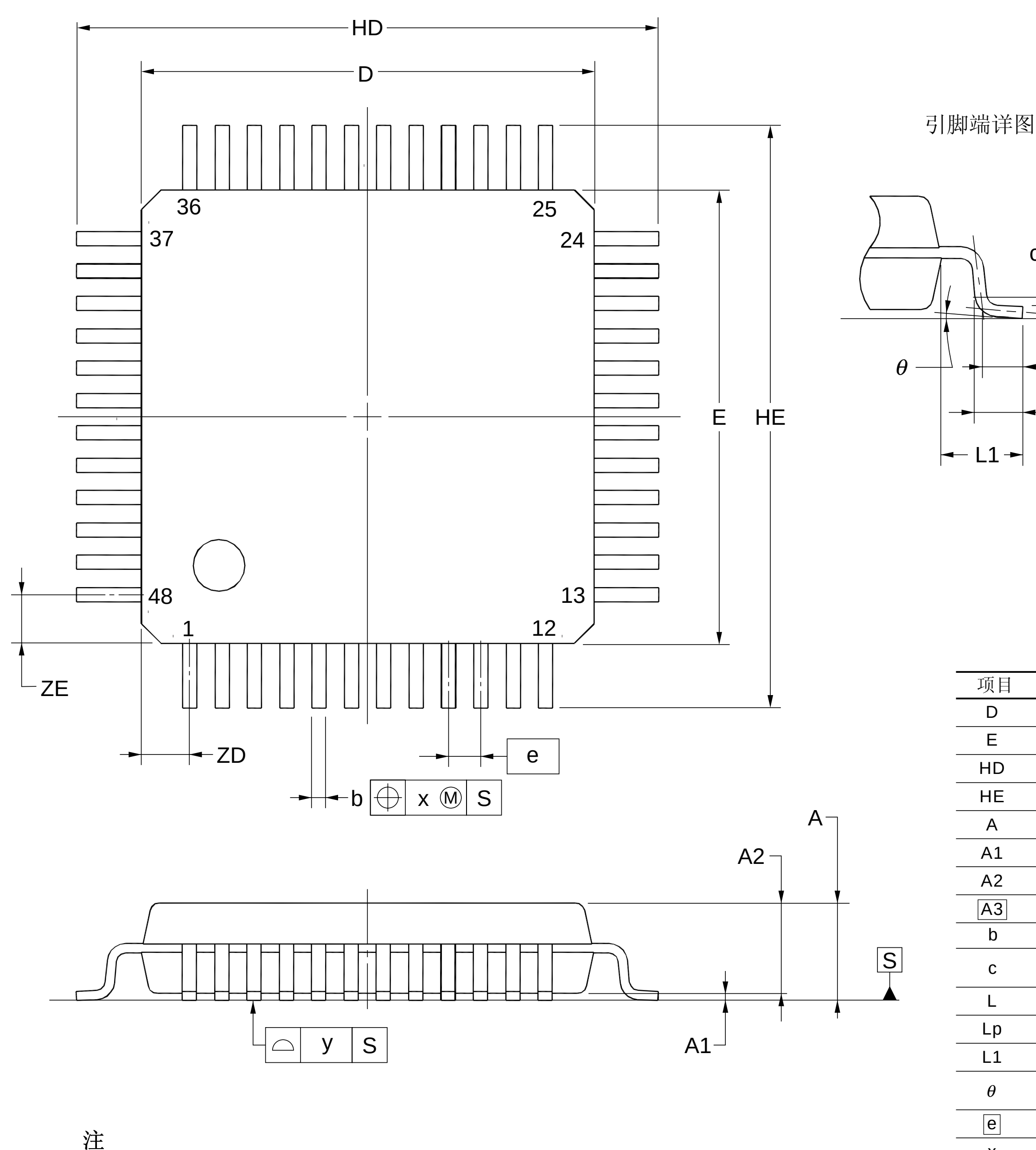
**P44GB-80-GAF**

注

该结构在最大使用用材料情况下，每条引脚的中心线位于其实际位置的 0.20mm 内

- **μPD78F0511GA-8EU-A, 78F0512GA-8EU-A, 78F0513GA-8EU-A, 78F0514GA-8EU-A, 78F0515GA-8EU-A, 78F0515DGA-8EU-A**

## 48 引脚塑封 LQFP(密脚距)(7×7)




(UNIT:mm)

| 项目 | 尺寸 |
|---|---|
| D | 7.00±0.20 |
| E | 7.00±0.20 |
| HD | 9.00±0.20 |
| HE | 9.00±0.20 |
| A | 1.60 MAX. |
| A1 | 0.10±0.05 |
| A2 | 1.40±0.05 |
| A3 | 0.25 |
| b | 0.22±0.05 |
| c | $0.145^{+0.055}_{-0.045}$ |
| L | 0.50 |
| Lp | 0.60±0.15 |
| L1 | 1.00±0.20 |
| θ | $3°^{+5°}_{-3°}$ |
| e | 0.50 |
| x | 0.08 |
| y | 0.08 |
| ZD | 0.75 |
| ZE | 0.75 |

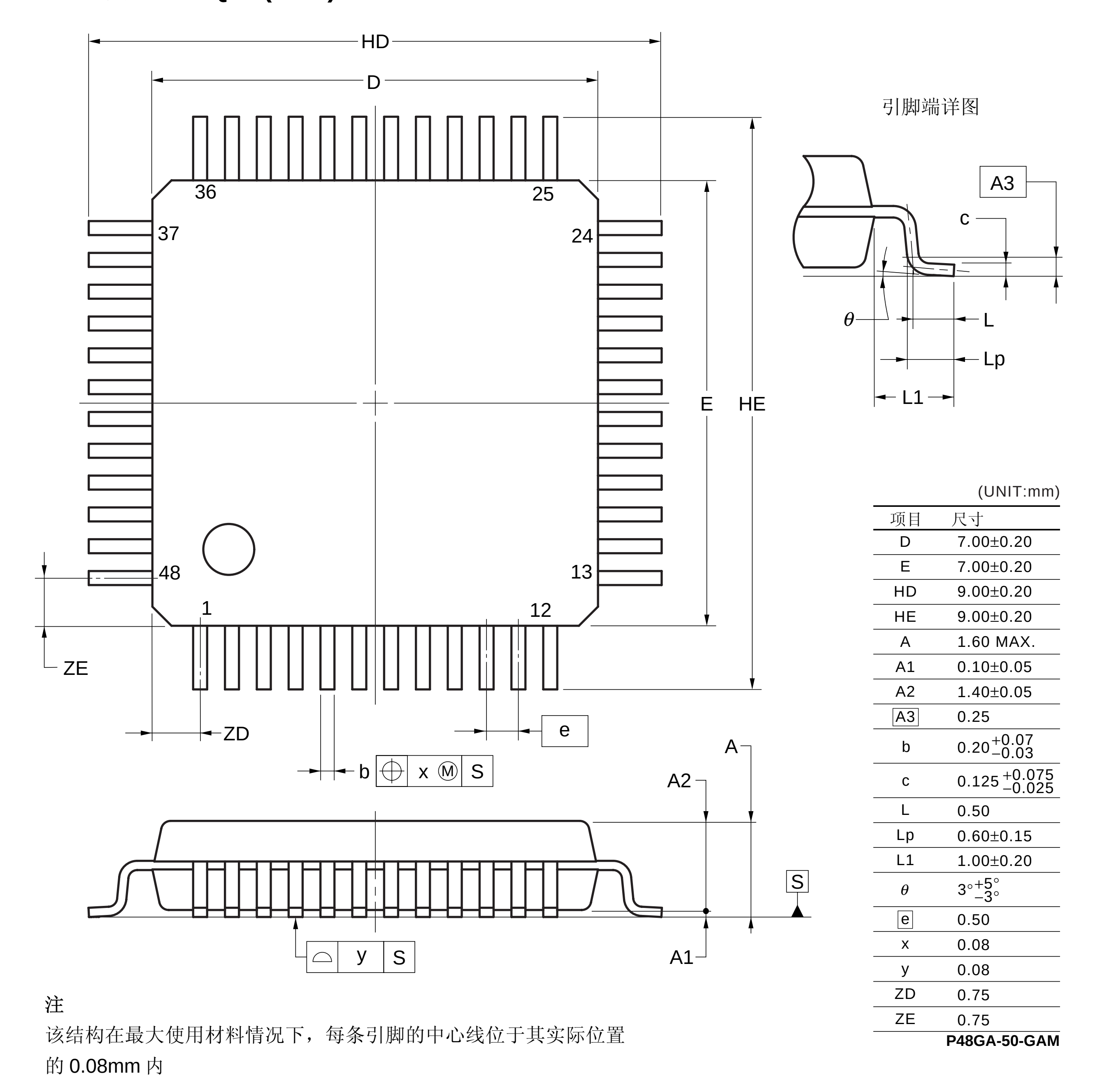
**P48GA-50-8EU**

注

该结构在最大使用材料情况下，每条引脚的中心线位于其实际位置的 0.08mm 内

<R> • **μPD78F0511GA(A)-GAM-AX, 78F0512GA(A)-GAM-AX, 78F0513GA(A)-GAM-AX, 78F0514GA(A)-GAM-AX, 78F0515GA(A)-GAM-AX**

## 48 引脚塑封 LQFP(7×7)




| 项目 | 尺寸 |
|---|---|
| D | 7.00±0.20 |
| E | 7.00±0.20 |
| HD | 9.00±0.20 |
| HE | 9.00±0.20 |
| A | 1.60 MAX. |
| A1 | 0.10±0.05 |
| A2 | 1.40±0.05 |
| A3 | 0.25 |
| b | $0.20^{+0.07}_{-0.03}$ |
| c | $0.125^{+0.075}_{-0.025}$ |
| L | 0.50 |
| Lp | 0.60±0.15 |
| L1 | 1.00±0.20 |
| θ | $3°^{+5°}_{-3°}$ |
| e | 0.50 |
| x | 0.08 |
| y | 0.08 |
| ZD | 0.75 |
| ZE | 0.75 |

注

该结构在最大使用材料情况下，每条引脚的中心线位于其实际位置的 0.08mm 内

<R>

# 第三十一章 推荐焊接条件

这些产品应在以下推荐条件下焊接和安装。
对于不同与以下推荐的焊接方法，请咨询 NEC 电子的销售代表。
对于技术信息，请参看以下网页。

半导体设备安装手册 (http://www.necel.com/pkg/en/mount/index.html)

**备注** 对于(A)级产品的焊接条件的评价是不完善的，因为其产品还在开发当中。

**表 31-1. 贴装型的焊接条件**

- **48 引脚塑封 LQFP (7 × 7)**
  μPD78F0511GA-8EU-A, 78F0512GA-8EU-A, 78F0513GA-8EU-A, 78F0514GA-8EU-A, 78F0515GA-8EU-A, 78F0515DGA-8EU-A[注 1]
- **80 引脚塑封 LQFP (12 × 12)**
  μPD78F0511GB-UES-A, 78F0512GB-UES-A, 78F0513GB-UES-A, 78F0513GB-UES-A[注 1]

| 焊接方法 | 焊接条件 | 推荐型号 |
|---|---|---|
| 红外加热 | 封装最高温度：260°C， 时间：最多 60 秒 （在 220°C或更高温度时），次数: 小于等于 3 次，暴露限制: 7 天[注 2] (之后在 125°C预烘 20 ～~ 72 小时) | IR60-207-3 |
| 局部加热 | 引脚温度： 350°C 最大， 时间： 3 秒 最大. (每排引脚) | – |

注 **1.** μPD78F0513D 和 78F0515D 具有片上调试功能。不要大规模使用这些产品，因为在片上调试功能使用后其可靠性不能保证，应考虑 Flash 存储器的写入次数。NEC 电子不接受与这些产品有关的投诉。
**2.** 在打开干燥的封装后，在存放时期内应将其放置于小于等于 25°C、 65% RH 的环境中。

**注意事项** 不要一起使用不同的焊接方法（除局部加热法以外）。

# 第三十二章　等待注意事项

## 32.1 等待注意事项

本产品有两种内部系统总线。

一种是 CPU 总线，另一种是连接低速外围硬件的外设总线。

由于 CPU 总线时钟与外设总线时钟是异步的，因此如果对 CPU 的访问与对外设的访问冲突时，可能会有不期望的非法数据产生。

因此当访问可能会产生冲突的外设时，CPU 反复执行处理，直至出现正确数据。

这样，CPU 并不执行下一条指令，而是等待。如果出现这种情况，指令执行所需的时钟数会以等待的时钟数而增加（如需了解等待时钟数的详细内容，参见**表 32-1**）。在进行实时操作时必须要注意这一点。

## 32.2 产生等待的外围硬件

表 32-1 列出了当被 CPU 访问时发出等待请求的寄存器和 CPU 等待的时钟数。

**表 32-1 产生等待的寄存器和 CPU 等待时钟数**

<table>
<tr><th>外围硬件</th><th>寄存器</th><th>访问方式</th><th>等待时钟的个数</th></tr>
<tr><td>串行接口<br>UART0</td><td>ASIS0</td><td>读</td><td>1 个时钟 (恒定)</td></tr>
<tr><td>串行接口<br>UART6</td><td>ASIS6</td><td>读</td><td>1 个时钟 (恒定)</td></tr>
<tr><td>串行接口 IIC0</td><td>IICS0</td><td>读</td><td>1 个时钟 (恒定)</td></tr>
<tr><td rowspan="5">A/D 转换器</td><td>ADM</td><td>写</td><td rowspan="4">1 ～ 5 个时钟 (当选择 $f_{AD} = f_{PRS}/2$ 时)<br>1 ～ 7 个时钟 (当选择 $f_{AD} = f_{PRS}/3$ 时)<br>1～ 9 个时钟 (当选择 $f_{AD} = f_{PRS}/4$ 时)<br>2 ～ 13 个时钟 (当选择 $f_{AD} = f_{PRS}/6$ 时)<br>2 ～ 17 个时钟 (当选择 $f_{AD} = f_{PRS}/8$ 时)<br>2 ～ 25 个时钟 (当选择 $f_{AD} = f_{PRS}/12$ 时)</td></tr>
<tr><td>ADS</td><td>写</td></tr>
<tr><td>ADPC</td><td>写</td></tr>
<tr><td>ADCR</td><td>读</td></tr>
<tr><td colspan="3">上述时钟个数是指在选择相同的源时钟作为 $f_{CPU}$ 和 $f_{PRS}$. 时的值。可以根据以下表达式和条件计算等待时钟的个数。<br><计算等待时钟数><br>• 等待时钟数 = $\frac{2 f_{CPU}}{f_{AD}}$ *<br>* 如果等待时钟的个数≤ 0.5 则将小数点以后的部分舍去，而如果等待时钟的个数> 0.5，则进行四舍五入<br>$f_{AD}$:　A/D 转换时钟的频率 ($f_{PRS}/2$ ～ $f_{PRS}/12$)<br>$f_{CPU}$:　CPU 时钟频率<br>$f_{PRS}$:　外围硬件时钟频率<br>$f_{XP}$:　主系统时钟频率<br><等待时钟的最大/最小个数所需的条件><br>• 最大个数: CPU 的最高速度($f_{XP}$)，A/D 转换时钟的最低速度($f_{PRS}/12$)<br>• 最小个数: CPU 的最低速度($f_{SUB}/2$)，A/D 转换时钟的最高速度 ($f_{PRS}/2$)</td></tr>
</table>

**注意事项　当 CPU 使用副系统时钟而外围硬件时钟停止时，不要使用产生等待请求的方式访问上述寄存器。**

**备注**　这里的时钟是指 CPU 时钟（$f_{CPU}$）。

# 附录 A　开发工具

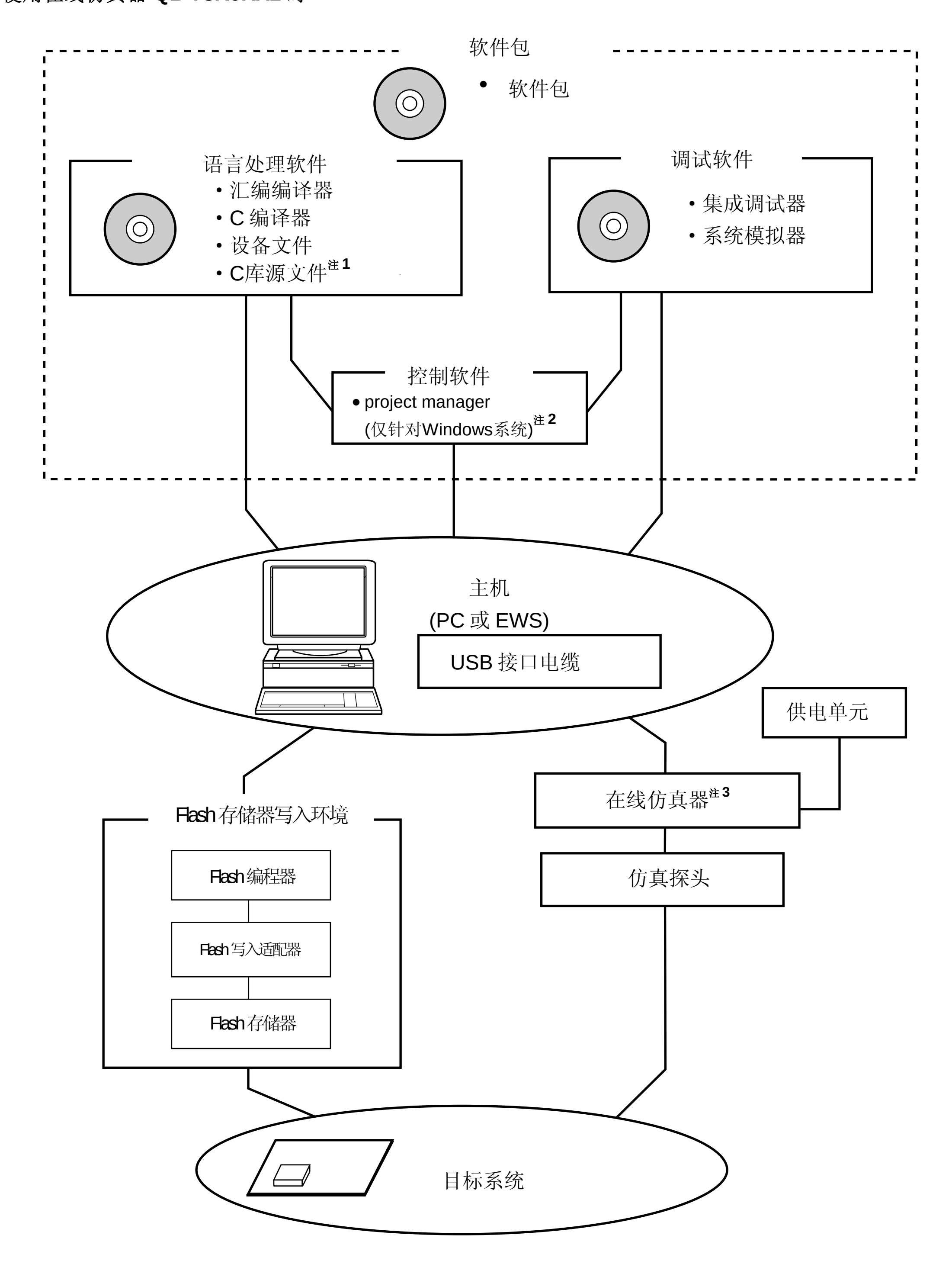
78K0/KC2 的系统开发中可以使用以下开发工具。
图 A-1 显示了开发工具的组成。

- **支持 PC98-NX 系列**
  除非特别说明，IBM PC/AT™ 兼容机支持的产品与PC98-NX系列计算机是兼容的。当使用PC98-NX 系列计算机时，参阅IBM PC/AT兼容机的使用说明。

- **Windows™**
  除非特别说明，“Windows” 是指以下几种操作系统。
  - Windows 98
  - Windows NT™
  - Windows 2000
  - Windows XP

图 **A-1** 开发工具的组成 **(1/2)**

**(1)** 当使用在线仿真器 **QB-78K0KX2** 时



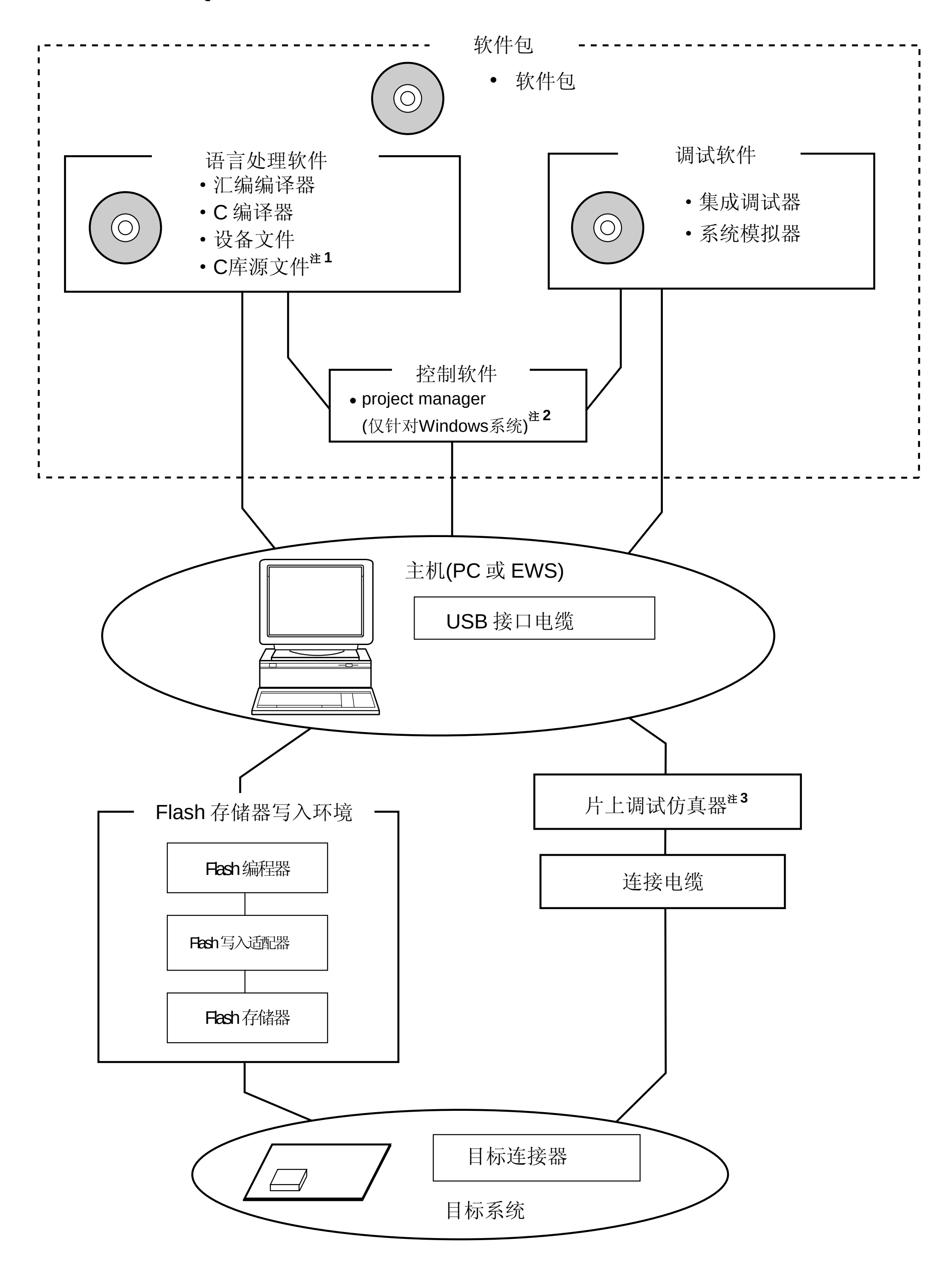

注 **1.** 软件包中不包括 C 库源文件。

**2.** 汇编编译器中包括 project manager PM+ 。
PM+仅用于 Windows。

**3.** 购买在线仿真器 QB-78K0KX2 时还提供集成调试器 ID78K0-QB、简单闪存编程器 PG-FPL3，电源和 USB 接口电缆。其它产品均独立销售。

图 **A-1.** 开发工具的组成**(2/2)**

★ **(2)** 当使用片上调试仿真器 **QB-78K0MINI** 时




注　**1.** 软件包中不包括 C 库源文件。

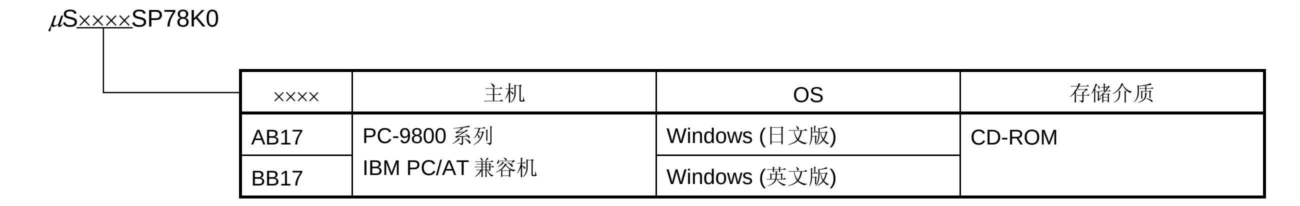
**2.** 汇编编译器中包括 project manager PM+ 。
PM+ 仅用于 Windows。

**3.** 购买片上调试仿真器 QB-78K0MINI 时还提供集成调试器 ID78K0-QB、USB 接口电缆和连接电缆。其它产品均独立销售。

## A.1 软件包

| SP78K0<br>78K/0 系列软件包 | 该软件包中的开发工具与 78k/0 系列相同。 |
|---|---|
| | 编号： μS××××SP78K0 |

**备注** 编号中的××××随主机和使用的 OS 而变化。



μS××××SP78K0

| ×××× | 主机 | OS | 存储介质 |
|---|---|---|---|
| AB17 | PC-9800 系列<br>IBM PC/AT 兼容机 | Windows (日文版) | CD-ROM |
| BB17 | | Windows (英文版) | |

## A.2 语言处理软件

| RA78K0<br>汇编编译器 | 该汇编编译器将助记符编写的程序转换为微处理器可执行的目标代码。<br>该汇编编译器可以自动建立符号表和对转移指令进行优化。<br>该汇编编译器应当和设备文件(DF780547) (独立销售)一起使用。<br>**<PC 环境中使用 RA78K0 的注意事项>**<br>该汇编编译器是基于 DOS 应用的。它也可以在 Windows 环境下使用，但必须在 Windows 环境下使用 program manager(包含在汇编编译器软件包中)。 |
|---|---|
| | 编号： μS××××RA78K0 |
| CC78K0<br>C 编译器 | 该编译器将 C 程序转换成微控制器可执行的目标代码。<br>该编译器应当和汇编程序、设备文件(二者均独立销售)结合使用。<br>**<PC 环境中使用 CC78K0 的注意事项>**<br>C 编译器是基于 DOS 应用的。它也可以在 Windows 环境下使用，但必须在 Windows 环境下使用 program manager (包含在汇编编译器软件包中)。 |
| | 编号： μS××××CC78K0 |
| DF780547 注 1<br>设备文件 | 该文件包含设备特有的信息。<br>该设备文件应当结合工具(RA78K0、CC78K0、用于 SM+ 的 78K0/KX2 和 ID78K0-QB) (均为独立销售)一起使用。<br>相应的 OS 和主机随使用工具而变化。 |
| | 编号：μS××××DF780547 |
| CC78K0-L 注 2<br>C 库源文件 | 这是一个函数的源文件，用来配置 C 编译器中的目标库。<br>该文件需要和用户使用规范说明中的目标库(包含在 C 编译器中)相匹配。 |
| | 编号： μS××××CC78K0-L |

注 **1.** DF780547 可以与 RA78K0、CC78K0 和 SM+ 一起用于 78K0/KX2 和 ID78K0-QB。

**2.** 软件包(SP78K0)不包括 CC78K0-L。

备注 编号中的××××随主机和使用的 OS 而变化。

μS××××RA78K0
μS××××CC78K0
μS××××CC78K0-L

| ×××× | 主机 | OS | 存储介质 |
|---|---|---|---|
| AB17 | PC-9800 系列<br>IBM PC/AT 兼容机 | Windows (日文版) | CD-ROM |
| BB17 | | Windows (英文版) | |
| 3P17 | HP9000 系列 700™ | HP-UX™ (Rel. 10.10) | |
| 3K17 | SPARC工作站™ | SunOS™ (Rel. 4.1.4)<br>Solaris™ (Rel. 2.5.1) | |

μS××××DF780547

| ×××× | 主机 | OS | 存储介质 |
|---|---|---|---|
| AB13 | PC-9800 系列<br>IBM PC/AT 兼容机 | Windows (日文版) | 3.5 英寸 2HD FD |
| BB13 | | Windows (英文版) | |

## A.3 控制软件

| PM+<br>project manager | 这是一个控制软件，可使用户在 Windows 环境下有效地进行软件开发。用户程序开发中进行的任何操作，如启动编辑器、构造程序和启动调试器，都可以由 project manager 执行。<br>**<注意事项>**<br>项目管理器包含在汇编编译器软件包 (RA78K0) 中。<br>仅在 Windows 中使用。 |
|---|---|

## A.4 Flash 存储器写入工具

| | |
|---|---|
| FL-PR4 , PG-FP4<br>Flash 存储器编程器 | Flash 存储器编程器，专用于内置 Flash 存储器的微控制器。 |
| <R> PG-FPL3 , FP-LITE3<br>简易 Flash 存储器编程器 | 简易 Flash 存储器编程器，专用于内置 Flash 存储器的微控制器。 |
| <R> FA-78F0515GA-8EU-MX<br><R> FA-78F0513GB-UES-MX<br>FA-44GB-8ES-A<br>Flash 存储器写入适配器 | Flash 存储器写入适配器用于连接 FlashPro4<br>• FA-78F0515GA-8EU-MX：<br>48 引脚塑封 LQFP (GA-8EU, GA-GAM 型)<br>• FA-78F0513GB-UES-MX：<br>44 引脚塑封 LQFP (GB-UES, GB-GAF 型) |

<R> 备注 **1.** FL-PR4, FP-LITE3, FA-78F0515GA-8EU-MX, FA-78F0513GB-UES-MX, 和 FA-44GB-8ES-A 是 Naito Densei Machida Mfg. Co., Ltd 的产品。

电话：+81-45-475-4191 Naito Densei Machida Mfg. Co., Ltd.

<R> **2.** 请使用 Flash 存储器写入适配器的最新产品。

## A.5 调试工具 (硬件)

### A.5.1 当使用在线仿真器 QB-78K0KX2 时

| 产品 | 说明 |
|---|---|
| QB-78K0KX2[注]<br>在线仿真器 | 在使用 78K0/Kx2 开发应用系统时，在线仿真器用于调试硬件和软件。它支持集成调试器(ID78K0-QB)。该仿真器应当与电源、仿真头及用于连接仿真器和主机的 USB 电缆结合使用。 |
| QB-144-CA-01<br>引脚检测适配器 | 该适配器用于示波器等的波形检测 |
| QB-80-EP-01T<br>仿真探头 | 这是一种可以灵活使用的仿真探头，用于连接在线仿真器和目标系统。 |
| QB-44GB-EA-03T,<br>QB-48GA-EA-02T<br>交换适配器 | 该适配器用于实现从在线仿真器到目标连接器的引脚转换<br>• QB-44GB-EA-03T: 44 引脚塑封 LQFP (GB-UES, GB-GAF 型)<br>• QB-48GA-EA-02T: 48 引脚塑封 LQFP (GA-8EU, GA-GAM 型) |
| QB-44GB-YS-01T,<br>QB-48GA-YS-01T<br>空间适配器 | 该适配器用于调整目标系统和在线仿真器之间的高度<br>• QB-44GB-YS-01T: 44 引脚塑封 LQFP (GB-UES, GB-GAF 型)<br>• QB-48GA-YS-01T: 48 引脚塑封 LQFP (GA-8EU, GA-GAM 型) |
| QB-44GB-YQ-01T,<br>QB-48GA-YQ-01T<br>YQ 连接器 | 该连接器用于连接目标连接器和交换适配器。<br>• QB-44GB-YQ-01T: 44 引脚塑封 LQFP (GB-UES, GB-GAF 型)<br>• QB-48GA-YQ-01T: 48 引脚塑封 LQFP (GA-8EU, GA-GAM 型) |
| QB-44GB-HQ-01T,<br>QB-48GA-HQ-01T<br>安装适配器 | 该适配器用于安装目标设备(通过插座安装到目标设备上)<br>• QB-44GB-HQ-01T: 44 引脚塑封 LQFP (GB-UES, GB-GAF 型)<br>• QB-48GA-HQ-01T: 48 引脚塑封 LQFP (GA-8EU, GA-GAM 型) |
| QB-44GB-NQ-01T,<br>QB-48GA-NQ-01T<br>目标连接器 | 该连接器用于将在线仿真器安装到目标系统上<br>• QB-44GB-NQ-01T: 44 引脚塑封 LQFP (GB-UES, GB-GAF 型)<br>• QB-48GA-NQ-01T: 48 引脚塑封 LQFP (GA-8EU, GA-GAM 型) |

注　使用 QB-78K0KX2 时应提供电源、USB 接口电缆。作为控制软件，还提供集成调试器 ID78K0-QB 和简易 Flash 存储器编程器 PG-FPL3。

**备注**　封装内容随产品编号而变化，如下所示：

| 封装内容 / 产品编号 | 在线仿真器 | 仿真头 | 交换适配器 | YQ 连接器 | 目标连接器 |
|---|---|---|---|---|---|
| QB-78K0KX2-ZZZ | QB-78K0KX2 | 无 | | | |
| QB-78K0KX2-T44GB | | QB-80-EP-01T | QB-44GB-EA-03T | QB-44GB-YQ-01T | QB-44GB-NQ-01T |
| QB-78K0KX2-T48GA | | | QB-48GA-EA-02T | QB-48GA-YQ-01T | QB-48GA-NQ-01T |

### ★ A.5.2 当使用片上调试仿真器 QB-78K0MINI 时

| 产品 | 说明 |
|---|---|
| QB-78K0MINI[注]<br>片上调试仿真器 | 在使用 78K0/Kx2 开发应用系统时，片上调试仿真器用于调试硬件和软件。它支持集成调试器(ID78K0-QB)。该仿真器使用一条连接电缆和一条用于连接主机的 USB 接口电缆。 |
| 目标连接器规范说明 | 10 引脚通用连接器(2.54 mm 脚距) |

注　使用 QB-78K0MINI 时提供 USB 接口电缆和一个连接电缆。作为控制软件，还提供集成调试器 ID78K0-QB。

## A.6 调试工具(软件)

| | |
|---|---|
| 用于 78K0/KX2 的 SM+<br>系统模拟器 | 用于 78K0/KX2 的 SM+是基于 Windows 的软件。<br>在主机上模拟目标系统操作时，该仿真器用来执行 C 源程序或汇编程序级的调试操作。<br>使用 78K0/KX2 的 SM+可在不依赖硬件开发的基础上进行应用逻辑测试和性能测试，从而提供了较高的开发效率和软件质量。<br>78K0/KX2 的 SM+应当结合设备文件(DF780547) (独立销售)使用。 |
| | 编号：SM780547-B |
| ID78K0-QB<br>集成调试器 | 该调试器支持 78K/0 系列的在线仿真器。ID78K0-QB 是基于 Windows 的软件。<br>它改善了 C 兼容的调试功能，并使用集成窗口功能(结合源程序、分开显示和存储区显示)来显示源程序的跟踪结果。该调试器应当结合设备文件(独立销售)一起使用。 |
| | 编号：μS××××ID78K0-QB |

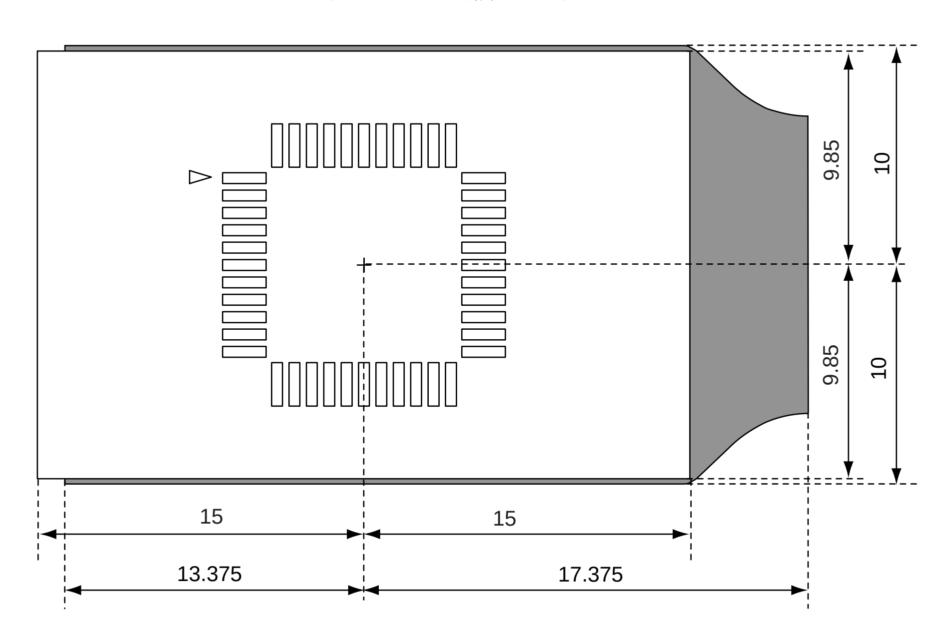
**备注**　　编号中的××××随主机和使用的 OS 不同而变化。

μS××××SM780547-B
μS××××ID78K0-QB

| ×××× | 主机 | OS | 存储介质 |
|---|---|---|---|
| AB17 | PC-9800 系列<br>IBM PC/AT 兼容机 | Windows (日文版) | CD-ROM |
| BB17 | | Windows (英文版) | |

本章显示了在使用 QB-78K0KX2 时，目标系统中禁止安装组件的区域和安装组件受严格限制的区域。

**图 B-1. 44 引脚 GB 封装**

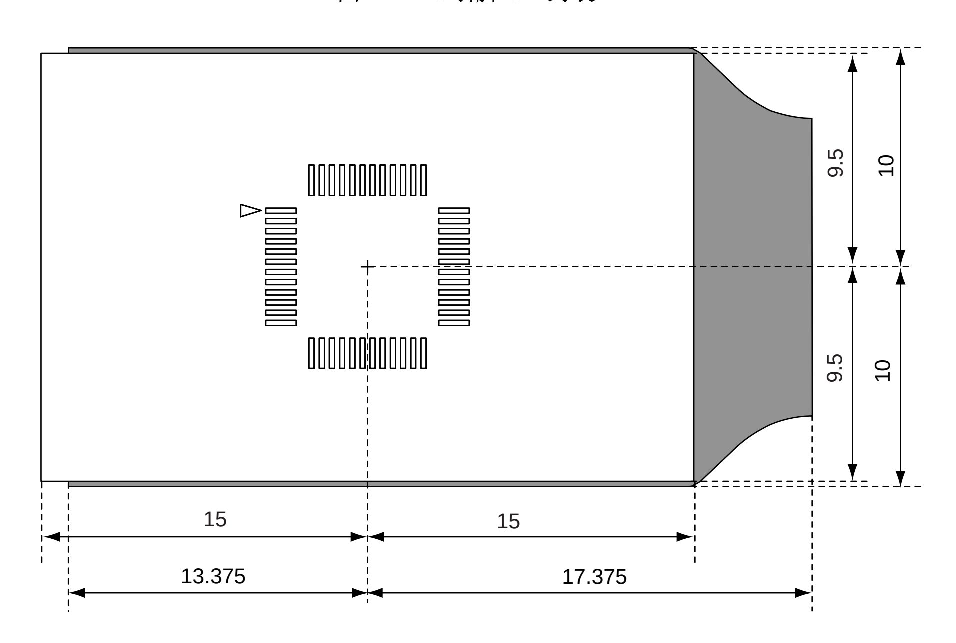



□: 交换适配器区域: 可以安装最高达 17.45 mm的组件[注]

■: 仿真器探头区域: 可以安装最高达 24.45 mm的组件[注]

**注** 使用空间适配器可以调整高度(每个增加 2.4 mm)

**图 B-2. 48 引脚 GA 封装**




□: 交换适配器区域: 可以安装最高达 17.45 mm的组件[注]

■: 仿真器探头区域: 可以安装最高达 24.45 mm的组件[注]

注 使用空间适配器可以调整高度(每个增加 2.4 mm)

附录 C 寄存器索引

## C.1 寄存器索引 (按寄存器名称的字母顺序排列)

**[A]**



**[B]**



**[C]**



**[E]**



**[I]**

## C.2 寄存器索引(按寄存器符号的字母顺序排列)

[W]

<R>

# 附录 D 注意事项列表

本附录列列出了本文档中所有的注意事项的描述。
在下表中有“分类（硬件/软件）”，如下。

硬件： 针对微控制器内部/外部硬件的注意事项。
软件： 针对软件，例如寄存器设置或者程序方面的注意事项。

(1/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第一章 | 硬件 | 引脚功能 | AVss | 使AVss与Vss的电位相同。 | p. 20, 22 ☐ |
| | | | REGC | 通过一个电容(0.47μF到1μF: 推荐)将REGC引脚和Vss连接到一起。 | p. 20, 22 ☐ |
| | | | ANI0/P20 ～ ANI7/P27 | 复位后，ANI0/P20 到 ANI7/P27设置为模拟输入模式。 | p. 20, 22 ☐ |
| 第二章 | 硬件 | 引脚功能 | ANI0/P20 ～ ANI7/P27 | 复位后，ANI0/P20 至 ANI7/P27被设置为模拟输入模式。 | p. 35 ☐ |
| | | | P31/INTP2/ OCD1A | 在使用具有片上调试功能的产品 (μPD78F0513D 和 78F0515D) 时，应确保在复位以前将引脚P31/INTP2/OCD1A[注] 下拉，防止故障发生。 | p. 35 ☐ |
| | | | | 对于具有48KB或更大Flash存储器而没有片上调试功能的产品 (μPD78F0514 和 78F0515)，有一个“I”, “K”, 或 “E” 的产品等级，对于具有片上调试功能的产品 (μPD78F0513D 和 78F0515D) 在使用Flash编程器写Flash存储器时，应按以下连接P31/INTP2/OCD1A[注]。<br>• P31/INTP2/OCD1A[注]: 通过电阻连接到 Vss (10 kΩ: 推荐)。<br>当通过自编程方式写入Flash存储器时，以上连接并不需要。 | p. 36 ☐ |
| | | | P121/X1/ OCD0A | 对于具有48KB或更大Flash存储器而没有片上调试功能的产品 (μPD78F0514 和 78F0515)，有一个“I”, “K”, 或 “E” 的产品等级，对于具有片上调试功能的产品 (μPD78F0513D 和 78F0515D) 在使用Flash编程器写Flash存储器时，应按以下连接P121/X1/OCD0A[注]。<br>• P121/X1/OCD0A[注]: 通过电阻连接到 Vss (10 kΩ: 推荐)(在输入模式)或悬空(在输出模式)。<br>当通过自编程方式写入Flash存储器时，以上连接并不需要。 | p. 38 ☐ |
| | | | REGC 引脚 | 在上图虚线部分，用尽可能短的配线连接。 | p. 39 ☐ |
| 第三章 | 软件 | 存储空间 | IMS, IXS:<br>内存容量切换寄存器，<br>内部扩展RAM容量切换寄存器 | 不管内存容量有多少，内存容量切换寄存器 (IMS) 和所有78K0/KC2产品的内部扩展RAM容量切换寄存器 (IXS) 的初始值都是固定的 (IMS = CFH, IXS = 0CH)。因此每种产品设置值如下所示。 | p. 44 ☐ |
| | | | SFR:<br>特殊功能寄存器 | 不要访问那些未分配特殊功能寄存器的地址区域。 | p. 56 ☐ |
| | | | SP:<br>堆栈指针 | 由于复位信号产生时，SP的内容不确定，所以在使用堆栈前必须先对SP初始化。 | p. 63 ☐ |
| 第四章 | 软件 | 软件功能 | P10/$\overline{\text{SCK10}}$/TxD0, P12/SO10 | 如果P10/$\overline{\text{SCK10}}$/TxD0和P12/SO10作为通用端口使用，则将串行操作模式寄存器10(CSIM10)和串行时钟选择寄存器10(CSIC10)设置为默认状态（00H）。 | p. 89 ☐ |
| | 硬件 | | 端口2 | 当端口2用于数字端口时，应使 AVREF 引脚与 VDD 引脚的电势相同。 | p. 94 ☐ |

(2/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 | |
|---|---|---|---|---|---|---|
| 第四章 | 硬件 | 软件功能 | P31/INTP2/ OCD1A | 在使用具有片上调试功能的产品 (μPD78F0513D 和 78F0515D) 时，应确保在复位以前将引脚P31/INTP2/OCD1A[注] 下拉，防止故障发生。 | p. 95 | □ |
| | | | | 对于具有48KB或更大Flash存储器而没有片上调试功能的产品 (μPD78F0514 和 78F0515)，有一个“I”, “K”, 或 “E” 的产品等级，对于具有片上调试功能的产品 (μPD78F0513D 和 78F0515D) 在使用Flash编程器写Flash存储器时，应按以下连接P31/INTP2/OCD1A[注]。<br>• P31/INTP2/OCD1A[注]: 通过电阻连接到 Vss (10 kΩ: 推荐)。<br>当通过自编程方式写入Flash存储器时，以上连接并不需要。 | p. 95 | □ |
| | 软件 | | P121/X1/OCD0A, P122/X2/EXCLK/ OCD0B, P123/XT1, P124/XT2/EXCLKS | 当使用P121 ～ P124引脚连接主系统时钟（X1,X2）或副时钟（XT1,XT2）的振荡器，或者输入该主系统时钟的外部时钟（EXCLK）或副时钟的外部时钟（EXCLKS）时，必须通过使用时钟操作模式选择寄存器（OSCCTL）对X1振荡模式、XT1振荡模式或者外部时钟输入模式进行设置（如需了解详细信息，可参见5.3 (1) 时钟操作模式选择寄存器（OSCCTL）和（3）副时钟引脚操作模式设置）。OSCCTL的复位值为00H（P121 ～ P124都用作I/O端口引脚）。在这种情况下，不必对PM121 ～ PM124和P121 ～ P124引脚进行设置。 | p. 102 | □ |
| | | | | 对于具有48KB或更大Flash存储器而没有片上调试功能的产品 (μPD78F0514 和 78F0515)，有一个“I”, “K”, 或 “E” 的产品等级，对于具有片上调试功能的产品 (μPD78F0513D 和 78F0515D) 在使用Flash编程器写Flash存储器时，应按以下连接P121/X1/OCD0A[注]。<br>• P121/X1/OCD0A[注]：P121/X1/OCD0A[注]: 通过电阻连接到 Vss (10 kΩ: 推荐)(在输入模式)或悬空(在输出模式)。<br>当通过自编程方式写入Flash存储器时，以上连接并不需要。 | p. 102 | □ |
| | | | 端口模式寄存器 | 对于44引脚产品，请设定PM0的2到7位、PM3的4到7位、PM4的2到7位、PM6的4到7位、PM7的4到7位和PM12的5到7位为1。<br>对于48引脚产品，请设定PM0的2到7位、PM3的4到7位、PM4的2到7位、PM6的4到7位、PM7的6到7位和PM12的1到7位为1。 | p. 108 | □ |
| | | | ADPC: A/D 端口配置寄存器 | 通过使用端口模式寄存器2(PM2)将A/D转换通道设置为输入模式。 | p. 111 | □ |
| | | | | 如果将数据写入ADPC，则产生一个等待周期。当CPU使用副系统时钟并且外部硬件时钟停止时，不要将数据写入ADPC。详细情况可参见 第三十二章 等待注意事项。 | p. 111 | □ |
| | | | 对端口寄存器n (Pn)的1位处理指令 | 当对一个支持输入输出功能的端口执行一个1位处理指令时，一个输入端口的输出锁存器的值（与目标位不同）可能被写入。<br><br>因此，建议一个端口从输入模式切换为输出模式时重写输出锁存器。 | p. 115 | □ |
| 第五章 | 软件 | 时钟发生器 | OSCSTL: 时钟操作模式选择寄存器 | 如果高速系统时钟振荡频率超过10MHz，则必须将AMPH设置为1。 | p. 120 | □ |
| | | | | 复位释放后外部功能设置之前应设置AMPH。复位释放后AMPH的值只能被修改一次。在AMPH被设置为1后， 当高速系统时钟（X1振荡）用作CPU时钟时，CPU时钟供应将停止4 .06～16.12μs。当高速系统时钟（外部时钟输入）用作CPU时钟时，CPU时钟在AMPH被设置为1后将停止160个外部时钟的时间。 | p. 120 | □ |

(3/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第五章 | 软件 | 时钟发生器 | OSCSTL:<br>时钟操作模式选择寄存器 | 当内部高速振荡时钟作为CPU时钟使用且AMPH=1时，执行STOP指令，则在STOP模式释放后，CPU时钟会停止4 .06～16.12μs，或者高速系统时钟（外部时钟输入）作为CPU时钟时，在STOP模式释放后CPU会停止160个外部时钟周期。当高速系统时钟（X1时钟）用作CPU时钟时，振荡稳定时间从STOP模式释放后开始计算。 | p. 120 □ |
| | | | | 若要修改EXCLK和OSCSEL，必须确保主OSC控制寄存器(MOC)的第7位(MSTOP) =1(X1振荡器停止或禁止使用来自EXCLK引脚的外部时钟)。 | p. 120 □ |
| | | | PCC:<br>处理器时钟控制寄存器 | 必须将第**3**位和第**7**位清零。 | p. 121 □ |
| | | | | 改变XTSTART、EXCLKS和OSCSELS的当前值时，应确保处理器时钟控制寄存器(PCC)的第5位(CLS) =0(CPU使用主系统时钟)。 | p. 122 □ |
| | | | RCM:<br>内部振荡模式寄存器 | 当设置RSTOP =1时，必须确保CPU使用的不是内部高速振荡时钟。特别地，在下列任一条件下，可以设置RSTOP =1。<br>• 当MCS = 1 (当CPU使用高速系统时钟时)<br>• 当CLS = 1 (当 CPU使用副系统时钟时)<br>此外，在将RSTOP设置为1之前应停止正在使用内部高速振荡时钟操作的外部硬件。 | p. 123 □ |
| | | | MOC:<br>主OSC控制寄存器 | 设置MSTOP = 1时，必须确保CPU使用的不是高速系统时钟。特别地，在下列任一条件下，可以设置MSTOP = 1。<br>• 当MCS = 0 (当CPU使用内部高速振荡时钟时)<br>• 当CLS = 1 (当CPU使用副系统时钟时)<br>此外，在设置MSTOP为1之前应停止正在使用高速系统时钟操作的外部硬件。 | p. 124 □ |
| | | | | 当时钟操作模式选择寄存器(OSCCTL)的第6位(OSCSEL)= 0(I/O端口模式)时，不要将MSTOP清零。 | p. 124 □ |
| | | | | 外部硬件时钟停止时外部硬件不能操作。外部硬件时钟停止后，若要恢复外部硬件的操作，则必须初始化外部硬件。 | p. 124 □ |
| | | | MCM:<br>主时钟模式寄存器 | XSEL在复位释放后只能被修改一次。 | p. 125 □ |
| | 硬件 | | | 无论XSEL和MCM0如何设置，除了fPRS以外的一个时钟将被提供给如下外围硬件功能<br>• 看门狗定时器(使用内部低速振荡时钟)<br>• 选择"$f_{RL}$", "$f_{RL}/2^7$", 或者 "$f_{RL}/2^9$"作为8位定时器H1的计数时钟(使用内部低速振荡时钟)<br>• 选择外部时钟作为外部硬件的时钟源<br>(以下情况除外： 选择TM0n (n = 0, 1)的外部计数时钟(TI00n 有效沿)时) | p. 125 □ |
| | 软件 | | OSTC:<br>振荡稳定时间计数器状态寄存器 | 在经过上述稳定时间后，从MOST11开始的位依次被设置为1，并一直保持。 | p. 126 □ |
| | | | | 振荡稳定时间计数器值增加到由OSTS设置的振荡稳定时间。当内部高速振荡时钟作为CPU时钟，如果已进入STOP模式，在释放该模式时，按如下方式设置振荡稳定时间。<br>• 预期的OSTC 振荡稳定时间 ≤ 由OSTS设置的振荡稳定时间。<br>注意，在STOP模式释放后，仅将达到振荡稳定时间(由OSTS设置)的状态赋给OSTC。 | p. 126 □ |
| | 硬件 | | | X1时钟振荡稳定等待时间不包括时钟振荡开始之前的时间(下图"a"表示的部分)。 | p. 126 □ |

(4/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第五章 | 软件 | 时钟发生器 | OSTS:<br>振荡稳定时间选择寄存器 | 当CPU使用X1时钟时，若要设置STOP模式，则必须在执行STOP指令之前设置OSTS。 | p. 127 □ |
| | | | | 在X1时钟振荡稳定时间内不要改变OSTS的值。 | p. 127 □ |
| | | | | 振荡稳定时间计数器值增加到由OSTS设置的振荡稳定时间。CPU使用内部高速振荡时钟时，如果已进入STOP模式，在释放该模式时，按如下方式设置振荡稳定时间。<br>• 预期的OSTC 振荡稳定时间 ≤ 由OSTS设置的振荡稳定时间。<br>注意，在STOP模式释放后，仅将达到振荡稳定时间的状态(由OSTS设置)赋给OSTC。 | p. 127 □ |
| | 硬件 | | | X1时钟振荡稳定等待时间不包括时钟振荡开始之前的时间(下图“a”表示的部分)。 | p. 127 □ |
| | | X1/XT1 振荡器 | – | 在使用X1振荡器和XT1振荡器时，图5-9和图5-10中被虚线包围的部分的配线应按照如下配线方法配线，以防止连接线电容产生不利影响。<br>• 连接线越短越好。<br>• 连接线不应与其他信号线交叉。流经的电流变化较大的信号线不要在振荡器周围布线。<br>• 要保持振荡器电容器的接地点电压与Vss相同。不要将电容的地信号接入大电流地。<br>• 不要从振荡器获取信号。<br>注意，XT1振荡器被设计成低振幅电路，以降低功耗。 | p. 129 □ |
| | | | | 当X2和XT1并行连接时，X2的串扰噪音会叠加到XT1，从而产生错误。 | p. 130 □ |
| | | 上电时的时钟发生器操作 | – | 在供电电压达到1.8V之前，如果电压上升斜率小于0.5 V/ms (MAX.)，则输入一个低电平到电源的RESET引脚直至电压达到1.8V，或者通过使用选项字节(POCMODE = 1)设置 2.7 V/1.59 V POC模式(见图5-13) 。通过这种方式，CPU的操作时序与 <2> 及由RESET引脚释放复位后的时序相同，如图5-12所示 。 | p. 134 □ |
| | | | | 当使用EXCLK和EXCLKS引脚的外部时钟输入时，不需要等待振荡稳定时间。 | pp. 134, 135 □ |
| | | | – | 当电源电压达到1.59 V (TYP.) 时，需要一个1.93 到 5.39 ms 的振荡稳定时间。如果在1.93 ms以内，电源电压从1.59 V (TYP.) 升到2.7 V (TYP.) , 则在复位处理以前，自动产生一个0 到 5.39 ms的振荡稳定时间。 | p. 135 □ |
| | | 高速系统时钟控制 | X1/P121, X2/EXCLK/P122 | 复位释放后，X1/P121和X2/EXCLK/P122 引脚在 I/O端口模式下使用。 | p. 136 □ |
| | | | X1 时钟 | 在X1时钟运行时，不要改变EXCLK和OSCSEL的值。 | p. 137 □ |
| | | | | 当供电电压达到所用时钟的操作电压后，设置X1时钟(参见　第二十八章　电气特性 (标准产品) 和 第二十九章　电气特性 ((A)等级产品，目标 ))。 | p. 137 □ |
| | | | 外部主系统时钟 | 在外部主系统时钟运行时，不要改变EXCLK和OSCSEL的值。 | p. 137 □ |
| | | | | 当供电电压达到所用时钟的操作电压后，设置外部主系统时钟(第二十八章　电气特性 (标准产品) 和 第二十九章　电气特性 ((A)等级产品，目标))。 | p. 137 □ |

(5/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第五章 | 软件 | 高速系统时钟控制 | 主系统时钟 | 如果选择高速系统时钟作为主系统时钟，则只有高速系统时钟可以被设置为外部硬件时钟。 | p. 138 ☐ |
| | | | 高速系统时钟 | 当MSTOP= 1时，必须确认MCS = 0 或 CLS = 1。此外，必须停止正在使用高速系统时钟的外部硬件。 | p. 139 ☐ |
| | | 内部高速振荡时钟控制 | 内部高速振荡时钟 | 当RSTOP= 1时，必须确认MCS = 1 或 CLS = 1。此外，必须停止正在使用内部高速振荡时钟的外部硬件。 | p. 141 ☐ |
| | | 副系统时钟控制 | XT1/P123, XT2/EXCLKS/P124 | 复位释放后XT1/P123和XT2/EXCLKS/P124引脚在I/O端口模式下使用。 | p. 141 ☐ |
| | | | XT1 时钟, 外部副系统时钟 | 在副系统时钟运行时，不要改变 XTSTART, EXCLKS和OSCSELS的值。 | p. 141 ☐ |
| | | | 副系统时钟 | 清零OSCSEL时，必须确认CLS = 0。此外，如果钟表定时器使用副系统时钟，则应停止其操作。 | p. 142 ☐ |
| | | | | 副系统时钟振荡不能通过使用STOP指令来停止。 | p. 142 ☐ |
| | | 内部低速振荡时钟控制 | 内部低速振荡时钟 | 如果通过选项字节选择“不能停止内部低速振荡器”，则不能控制内部低速振荡时钟的振荡。 | p. 143 ☐ |
| | | CPU 时钟 | – | 供电电压达到所用时钟的操作电压后，设置时钟(参见 第二十八章 电气特性(标准产品) 和 第二十九章 电气特性 ((A)等级产品，目标)) 。 | pp. 145, 146, 148 ☐ |
| | | | | 主系统时钟周期分频因子(PCC0～PCC2)的选择，以及从主系统时钟到副系统时钟的切换(CSS从0变到1) 不能同时设置。<br>但对于主系统时钟周期分频因子(PCC0～PCC2)的选择，以及副系统时钟到主系统时钟的切换(CSS从1变到0)可以同时设置。 | p. 150 ☐ |
| | | | | 当内部高速振荡时钟切换到高速系统时钟时，MCM的第2位(XSEL)必须预先被设置为1。XSEL值在复位释放后只能被修改一次。 | p. 151 ☐ |
| 第六章 | 硬件 | 16位定时器/事件计数器00 | – | TI010的有效沿和定时器输出 (TO00) 不能够被 P01 引脚同时使用。可任选其中一个功能。 | p. 154 ☐ |
| | 软件 | | | 如果16位定时器模式控制寄存器00(TMC00)的第2和3位 2 (TMC002 和 TMC003) 清零为00，并且输入触发信号，则捕捉数据不确定。 | p. 155 ☐ |
| | | | | 为了将捕捉模式改变为比较模式，首先要将 TMC003 和 TMC002 位清零 00，然后改变设置。<br>除非设备复位否则捕捉的值将一直保存在CR000中。如果已经改变为比较模式，请确保设定比较值。 | p. 155 ☐ |
| | | | TM00:<br>16位定时器计数器00 | 即使读取了TM00，也不能通过CR010捕捉到该值。 | p. 155 ☐ |
| | | | | 读取TM00时，暂停计数时钟输入并在读取定时器后再恢复操作。因此不会出现时钟丢失。 | p. 155 ☐ |
| | | | CR000, CR010: 16位定时器捕捉/比较寄存器000, 010 | 当CR000在比较模式时，即使输入触发信号，它也不会执行捕捉操作。 | p. 156 ☐ |
| | | | | 当CR010在比较模式时，即使输入触发信号，它也不会执行捕捉操作。 | p. 156 ☐ |

(6/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第六章 | 软件 | 16位定时器/事件计数器00 | CR000, CR010: 16位定时器捕捉/比较寄存器000, 010 | 如果要使用输入到TI000引脚的反向脉冲沿来捕捉TM00的计数值，并存入CR000，则在捕捉了计数值后，不产生中断请求信号(INTTM000)。如果在操作期间检测到TI010的有效沿，则不执行捕捉操作，但会产生INTTM000信号作为外部中断信号。当不使用外部中断时，可以屏蔽INTTM000信号。 | p. 158 ☐ |
| | | | TMC00：16位定时器模式控制寄存器 00 | 当设置TMC002与 TMC003为00（操作停止模式）以外的值时，16位定时器计数器00（TM00）开始计数。若要停止操作，则设置TMC002和 TMC003为00。 | p. 159 ☐ |
| | 硬件 | | CRC000: 捕捉/比较控制寄存器 00 | 为确保捕捉操作正确执行，捕捉触发需要的脉冲应大于两个计数时钟的时钟周期，该计数时钟用预分频器模式寄存器00 (PRM00)选择。 | p. 161 ☐ |
| | 软件 | | TOC00: 16位定时器输出控制寄存器 00 | 必须按以下步骤设置 TOC00。<br><1> 设置TOC004与TOC001为1。<br><2> 仅设置TOE00=1。<br><3> 设置LVS00或LVR00为1。 | p. 162 ☐ |
| | | | PRM00: 预分频器模式寄存器 00 | 当PRM001与PRM000为11时(指定TI000引脚的有效沿作为计数时钟)，不要进行以下设置。<br>• 清零&启动模式(通过TI000引脚有效沿进入)<br>• 设置TI000引脚作为捕捉触发 | p. 164 ☐ |
| | | | | 如果在TI000或TI010引脚为高电平并且指定TI000或TI010引脚的有效沿为上升沿或兼有两种脉冲沿时允许16位定时器/事件计数器00操作，TI000或TI010引脚的高电平将被检测为上升沿。当TI000或TI010引脚被上拉时要注意。然而一旦定时器操作已经被停止然后再次允许时则不检测上升沿。 | p. 164 ☐ |
| | 硬件 | | | 对于P01引脚，TI010的有效沿和定时器输出(TO00)不能同时使用，对于P06引脚，TI011的有效沿和定时器输出(TO01)也不能同时使用。可以任选其一。 | p. 164 ☐ |
| | 软件 | | 清零&启动模式(通过TI000引脚有效沿输入进入) | 不要将TI000 引脚的有效沿(PRM001与PRM000 = 11)设置为计数时钟。当PRM001与PRM000 = 11时，将TM00 清零。 | p. 175 ☐ |
| | | | PPG 输出 | 若在操作期间修改占空比（CR010寄存器的值），可参阅 6.5.1 TM00 操作期间CR010的重写。 | p. 197 ☐ |
| | | | | 对CR000与CR010进行设置，以满足条件 0000H ≤ CR010 < CR000 ≤ FFFFH 。 | p. 198 ☐ |
| | | | 单脉冲输出 | 当单脉冲输出时，不要再次输入触发(设置OSPT00=1或者检测TI000引脚的有效沿)。在当前单脉冲输出完成后产生触发，才可再次输出单脉冲。 | p. 200 ☐ |
| | | | | 如果要使用单脉冲输出触发(仅设置OSPT00=1)，则不要修改TI000引脚或其复用功能端口引脚的电平。否则可能会产生异常输出的脉冲。 | p. 200 ☐ |
| | | | | 不要为CR000与CR001设置相同的值。 | p. 202 ☐ |
| | | | LVS00, LVRn0 | 必须按上述<1>、<2>和<3>步骤设置LVS00和LVR00。<br>步骤<2>可以在<1>之后<3>之前执行。 | p. 214 ☐ |
| | | | – | 表 6-3显示了每个通道的限制条件。 | p. 215 ☐ |
| | 硬件 | | 定时器启动误差 | 定时器启动后，在产生相等信号之前可能发生将近一个时钟周期的误差。这是由于启动TM0n计数操作与计数脉冲不同步引起的。 | p. 215 ☐ |
| | 软件 | | CR000, CR010: 16位定时器捕捉/比较寄存器000, 010 | 为CR00n与CR01n设置0000H以外的值 (当TM0n用作外部事件计数器时不能进行单脉冲计数)。 | p. 215 ☐ |

(7/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第六章 | 软件 | 16位定时器/事件计数器00 | CR000, CR010: 16位定时器捕捉/比较寄存器000, 010 | 如果在读取CR000/CR010时，有效沿输入到TI000/TI010引脚并且检测到TI000引脚脉冲的反向沿时，CR010执行捕捉操作但CR000/CR010的读取值不确定。此时当检测到TI000/TI010引脚的有效沿时产生中断信号(INTTM000/INTTM010)(当检测到TI000引脚的反向脉冲沿时不产生中断信号)。<br>如果由于检测到TI000/TI010引脚的有效沿而捕捉到计数值时，则应在产生INTTM000/INTTM010信号后读取CR000/CR010的值。 | p. 216 ☐ |
| | | | | 当16位定时器/事件计数器00停止操作后，CR000与CR010的值不确定。 | p. 216 ☐ |
| | | | ES000, ES001 | 当定时器停止操作时(TMC003与TMC002 = 00)设置TI000 引脚的有效沿。使用ES000与ES001设置有效沿。 | p. 216 ☐ |
| | | | 再次触发单脉冲 | 必须确定在单脉冲输出模式下正在输出一个活动电平时，没有触发产生。而且必须在输出当前活动电平后输入下一个触发。 | p. 216 ☐ |
| | | | OVF00 | 在以下情况下以及TM00溢出时将OVF00 标志设置为1。<br>选择清零&启动模式(在TM00与CR000匹配时进入)。<br>→ 设置CR000 = FFFFH.<br>→ 当TM00与CR000匹配且TM00从FFFFH变为0000H时TM00被清零 | p. 217 ☐ |
| | | | | 即使在TM00 溢出后下一个计数时钟被计数之前(TM00的值变为0001H之前)将OVF00标志清零(0)，OVF00也会再次被设置为1，并且对其清零是无效的。 | p. 217 ☐ |
| | | | 单脉冲输出 | 在自由运行定时器模式下或者在清零&启动模式(通过TI000引脚有效沿进入)下，单脉冲输出能正确操作。而在清零&启动模式(在TM00与CR000匹配时进入)下不能输出单脉冲。 | p. 217 ☐ |
| | | | TI000 | 当指定TI000 的有效沿作为计数时钟时，捕捉寄存器(指定TI000作为触发)不能正确操作。 | p. 218 ☐ |
| | 硬件 | | TI000, TI010 | 为了准确捕捉计数值，作为捕捉触发的TI000与TI010引脚的输入脉冲宽度必须大于两个计数时钟(由PRM00选择)(参见图 **6-7**)。 | p. 218 ☐ |
| | | | INTTM000, INTTM010 | 在计数时钟的下降沿进行捕捉操作，但中断信号(INTTM000与INTTM010)则在下一个计数时钟的上升沿产生。(参见图 **6-7**)。 | p. 218 ☐ |
| | 软件 | | CRC001 = 1 | 在TI000引脚输入信号的反向沿将TM00的计数值捕捉到CR000时，捕捉到计数值后不产生中断信号(INTTM000)。如果在此操作期间，从TI010引脚检测到有效沿，则不执行捕捉操作但会产生中断信号(INTTM000)作为外部中断信号。当不使用外部中断时屏蔽INTTM000信号。 | p. 218 ☐ |
| | 硬件 | | 复位后指定有效沿 | 如果复位后TI000或TI010引脚处于高电平，且当指定上升沿或兼有两种脉冲沿作为TI000或TI010引脚的有效沿时，允许16位定时器/事件计数器00的操作，则检测到TI000或TI010引脚的高电平作为上升沿。上拉TI000或TI010引脚时要特别注意。但如果操作停止之后又允许操作，则不检测上升沿。 | p. 218 ☐ |
| | | | 采样时钟(用于消除噪音) | 根据TI000的有效沿是用作计数时钟还是用作捕捉触发，采样时钟(用于消除噪音)会发生变化。在前一种情况中采样时钟恒为$f_{PRS}$。而在后一种情况中则将由PRM00选择的计数时钟用于采样。<br>当采样到TI000引脚的输入信号并且在一个周期内两次检测到有效电平时，才认为检测到有效沿。因此可以消除窄脉冲宽度的噪音(参见图 **6-7**)。 | p. 218 ☐ |
| | | | TI000/TI010 | 无论CPU处于何种操作模式，当定时器停止操作时，不响应TI000/TI010引脚的输入信号。 | p. 218 ☐ |

(8/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第七章 | 软件 | 8位定时器/事件计数器50和51 | CR5n:<br>8位定时器比较寄存器5n | 在TM5n与CR5n (TMC5n6 = 0)匹配时清零和启动模式下，不要将其它值写入CR5n。 | p. 221 □ |
| | | | | 在PWM模式中，以至少3个计数时钟(由TCL5n选择时钟)的间隔对CR5n进行重写。 | p. 221 □ |
| | | | TCL50:<br>定时器时钟选择寄存器50 | 如果要向TCL50写入其它值，则必须提前停止定时器操作。 | p. 222 □ |
| | | | | 必须将第3 ～ 7位清零。 | p. 222 □ |
| | | | TCL51:<br>定时器时钟选择寄存器51 | 如果要向TCL51写入其它值，则必须提前停止定时器操作。 | p. 223 □ |
| | | | | 必须将第3 ～ 7位清零。 | p. 223 □ |
| | | | TMC5n:<br>8位定时器模式控制寄存器51 (TMC51) | 除PWM模式外，对LVS5n与LVR5n的设置有效。 | p. 225 □ |
| | | | | 按下列顺序执行<1> ～ <4>，注意不能同时执行。<br><1>　设置TMC5n1，TMC5n6:　操作模式设置<br><2>　设置TOE5n，允许输出:　允许定时器输出<br><3>　设置 LVS5n, LVR5n (见注意事项1):　定时器F/F设置<br><4>　设置TCE5n | p. 225 □ |
| | | | | 当 TCE5n = 1 时，禁止设定 TMC5n 的其它位。 | p. 225 □ |
| | | | 间隔定时器 | 操作期间不要向CR5n写入其它值。 | p. 227 □ |
| | | | 方波输出 | 操作期间不要向CR5n写入其它值。 | p. 230 □ |
| | | | PWM 输出 | 在PWM模式中，当对CR5n进行重写时，需要至少3个计数时钟(该计数时钟由TCL5n选择)周期的写间隔。 | p. 231 □ |
| | | | | 图7-15<1>与<2>之间读取CR5n时，读取的值与实际值不同(读取的值：M，CR5n实际值：N)。 | p. 234 □ |
| | | | 定时器启动误差 | 定时器启动后，在产生匹配信号之前可能发生将近一个时钟周期的误差。<br>这是由于8位定时器计数器50和51(TM50, TM51)的启动与计数时钟不同步引起的。 | p. 235 □ |
| 第八章 | 软件 | 8位定时器H0 和 H1 | CMP0n:<br>8位定时器H比较寄存器 0n (CMP0n) | CMP0n在定时器计数操作期间不能被重写。（只能更新与原数据相同的值）。 | p. 239 □ |
| | | | CMP1n:<br>8位定时器 H比较寄存器1n (CMP1n) | 在PWM输出模式和载波发生器模式下，当停止定时器计数操作(TMHEn = 0)后再启动定时器计数操作(TMHEn = 1)时，必须设置CMP1n（即使对CMP1n设置相同的值也要再设置一次）。 | p. 239 □ |
| | | | TMHMD0:<br>8位定时器H模式寄存器0 | 当TMHE0=1时，禁止设置TMHMD0的其他位。但是可以将TMHMD0更新（相同值）。 | p. 242 □ |
| | | | | 在PWM输出模式中，当定时器停止计数后（TMHE0 = 0）再次启动计数操作时（TMHE0 = 1），必须设置8位定时器H比较寄存器10（CMP10）的值（即使设置的是相同的值，也必须再次设置）。 | p. 242 □ |
| | | | TMHMD1:<br>8位定时器H模式寄存器1 | 当TMHE1=1时，禁止设置TMHMD1的其他位。但是可以将TMHMD1更新（相同值）。 | p. 243 □ |
| | | | | 在PWM输出模式和载波发生模式中，当定时器停止计数后（TMHE=0）再次启动计数器操作时（TMHE1=1）,必须设置8位定时器H比较寄存器11（CMP11）的值（即使设置的是相同的值，也必须再次设置）。 | p. 243 □ |
| | | | | 当使用载波发生器模式时，设置TMH1的计数时钟频率至少是TM51计数时钟频率的6倍。 | p. 243 □ |
| | | | TMCYC1:<br>8位定时器H载波控制寄存器1 | 当TMHE=1时，禁止重写RMC1。但是可以将TMCYC1更新（相同值）。 | p. 244 □ |

(9/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第八章 | 硬件 | 8位定时器H0 和 H1 | PWM 输出 | 可以在定时器计数操作时改变CMP1n的设置值。但从改变CMP1n寄存器的值到这个值被传输给寄存器需要经历三个操作时钟（通过TMHMDn寄存器的CKSn2 到CKSn0位选择的时钟信号） | p. 249 □ |
| | 软件 | | | 在定时器计数操作停止后（TMHEn=0），再次启动定时器操作时（TMHEn=1）,必须对CMP1n进行设置（即使是设置相同的值，也必须重新设置CMP1n寄存器） | p. 249 □ |
| | | | | CMP1n寄存器的设置值（M）和CMP0n寄存器的设置值（N）必须在以下范围内取值。<br>00H ≤ CMP1n (M) < CMP0n (N) ≤ FFH | p. 249 □ |
| | | | 载波发生器操作（仅用于8位定时器H1） | NRZB1位被重写后至少到第2个时钟时才能被再次重写，否则，从NRZB1位到NRZ1位的传送不能得到保证。. | p. 255 □ |
| | | | | 在载波发生器模式中使用8位定时器/事件计数器51时，在<1>时刻将产生中断。如果在其他模式中使用8位定时器/事件计数器51，中断产生的时序则不相同。 | p. 255 □ |
| | | | | 在定时器计数操作停止后（TMHE1=0），再启动定时器操作（TMHE1=1）时，必须设置CMP11（即使是相同的值，也必须对CMP11重新设置） | p. 257 □ |
| | | | | 设置TMH1的计数时钟频率至少是TM51的计数时钟频率的6倍。 | p. 257 □ |
| | | | | CMP01和CMP11的值必须在01H和FFH之间。 | p. 257 □ |
| | | | | 定时器计数期间可以改变CMP11寄存器的值。但是，从改变CMP11的值到将该值传输到寄存器需要三个操作时钟（由TMHMD1寄存器的CKS12～CKS10位选择的时钟信号） | p. 257 □ |
| | | | | 在计时操作开始前必须设置RMC1。 | p. 257 □ |
| 第九章 | 软件 | 钟表定时器 | WTM:<br>钟表定时器操作模式寄存器 | 在钟表定时器操作期间不要修改计数时钟和间隔时间 (通过设置WTM的第4 ～ 7位(WTM4 ～ WTM7)实现)。 | p. 264 □ |
| | 硬件 | | 中断请求 | 通过钟表定时器模式控制寄存器(WTM)(WTM的第0位(WTM0)和第1位(WTM1)置1)允许钟表定时器和5位计数器操作时，从寄存器被设置后到第1个中断请求(INTWT)产生的这段间隔时间与WTM的第2和3位(WTM2和WTM3)指定的间隔时间并不完全匹配。但其后的INTWT信号会以指定的时间间隔产生。 | p. 266 □ |
| 第十章 | 软件 | 看门狗定时器 | WDTE:<br>看门狗定时器允许寄存器 | 如果写入WDTE的值不是ACH，将产生一个内部复位信号。如果看门狗定时器的时钟源被停止，则当看门狗定时器的时钟源恢复操作时，将产生内部复位信号。 | p. 269 □ |
| | | | | 如果使用1位存储器操作指令对WDTE进行操作，将产生一个内部复位信号。如果看门狗定时器的时钟源被停止，当提供给看门狗定时器的时钟源恢复操作时，将产生内部复位信号。 | p. 269 □ |
| | | | | 从WDTE读取的值为9AH/1AH（与写入值（ACH）不同）。 | p. 269 □ |
| | | | 操作控制 | 复位释放后对WDTE进行第一次写操作，将清零看门狗定时器，如果这一操作是在到达溢出时间之前进行的而不管写操作时序，看门狗定时器再次开始计数。 | p. 270 □ |
| | | | | 如果通过将“ACH”写入WDTE来对看门狗定时器清零，则实际的溢出时间可能与由选项字节设置的溢出时间不同，最高可达$2/f_{RL}$秒。 | p. 270 □ |
| | | | | 在计数值溢出(FFFFH)之前，可将看门狗定时器立即清零。 | p. 270 □ |

(10/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第十章 | 软件 | 看门狗定时器 | 操作控制 | 如果LSROSC = 0，则在HALT或STOP模式释放后看门狗定时器恢复计数。此时，不对计数器清零，而从计数器停止时的值开始计数。 | p. 271 □ |
| | | | | 如果在LSROSC = 0时通过设置LSRSTOP(内部振荡模式寄存器(RCM)的第1位=1)停止内部低速振荡器的振荡，则看门狗定时器停止操作。此时不对计数器清零(0)。 | |
| | | | | 在flash存储器自编程和EEPROM™仿真期间看门狗定时器不停止操作。在处理期间中断响应被延迟。设置溢出时间和窗口大小时，应考虑延迟。 | |
| | | | 设置看门狗定时器溢出时间，设置看门狗定时器窗口打开周期 | 禁止出现WDCS2 = WDCS1 = WDCS0 = 0且WINDOW1 = WINDOW0 = 0的情况。 | pp. 271, 272 □ |
| | | | | 在flash存储器自编程和EEPROM仿真期间看门狗定时器不停止操作。在处理期间中断响应被延迟。设置溢出时间和窗口大小时，应考虑延迟。 | pp. 271, 272 □ |
| | | | 设置看门狗定时器窗口打开周期 | 复位释放后对WDTE进行第一次写操作，将清零看门狗定时器，如果这一操作是在到达溢出时间之前进行的而不管写操作时序，看门狗定时器再次开始计数。 | p. 272 □ |
| 第十一章 | 软件 | 时钟输出控制器 (仅48引脚产品) | CKS:<br>时钟输出选择寄存器 | 当时钟输出操作停止时(CLOE = 0)，设置CCS3 ～ CCS0。 | p. 276 □ |
| 第十二章 | 软件 | A/D转换器 | ADCR:<br>10位 A/D转换寄存器,<br>ADCRH:<br>8位 A/D转换寄存器 | 当从ADCR和ADCRH读取数据时，会产生一个等待周期。当CPU工作在副系统时钟，且外围硬件时钟处于停止状态时，不要从ADCR和ADCRH读取数据。需要了解详细信息，可参阅 第三十二章 等待注意事项。 | p. 280 □ |
| | | | ADM:<br>A/D转换器模式寄存器 | 在将FR0 ～ FR2位，LV1和LV0位修改成其他值之前，A/D转换必须停止。 | p. 282 □ |
| | | | | 如果对ADM赋值，会产生1个等待周期。当CPU工作在副系统时钟，且外围硬件时钟处于停止状态时，不要对ADM赋值。需要了解详细信息，可参阅 第三十二章 等待注意事项。 | p. 282 □ |
| | | | A/D 转换时间选择 | 根据以下条件设置转换时间<br>• $4.0\ V \le AV_{REF} \le 5.5\ V$: $f_{AD}$ = 0.6 ～ 3.6 MHz<br>• $2.7\ V \le AV_{REF} < 4.0\ V$: $f_{AD}$ = 0.6 ～ 1.8 MHz<br>• $2.3\ V \le AV_{REF} < 2.7\ V$: $f_{AD}$ = 0.6 ～ 1.48 MHz | p. 283 □ |
| | | | | 当修改FR2～FR0，LV1和LV0为其他值时，要预先停止A/D转换一次 (ADCS = 0) 。 | p. 283 □ |
| | | | | 当$2.3\ V \le AV_{REF} < 2.7\ V$时，修改LV1和LV0的默认值。 | p. 283 □ |
| | | | | 上述的转换时间不包括时钟频率误差。在选择转换时间的时候，要将时钟频率误差考虑进去。 | p. 283 □ |
| | | | ADCR:<br>10位 A/D转换寄存器 | 在对A/D转换器模式寄存器(ADM)和模拟输入通道选择寄存器(ADS)和A/D端口配置寄存器(ADPC)赋值时，ADCR的内容可能是不确定的。转换结束后先读取转换结果，再对ADM，ADS和ADPC赋值。如果没有按照上述时序，可能会读取到不正确的转换结果。 | p. 284 □ |
| | | | | 如果从ADCR读取数据，会产生一个等待周期。当CPU工作在副系统时钟，且外围硬件时钟处于停止状态时，不要从ADCR读取数据。需要了解详细信息，可参阅 第三十二章 等待注意事项 。 | p. 284 □ |

(11/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第十二章 | 软件 | A/D转换器 | ADCRH:<br>8位 A/D转换寄存器 | 在对A/D转换器模式寄存器(ADM)、模拟输入通道选择寄存器(ADS)和A/D端口配置寄存器(ADPC)赋值时, ADCRH的内容可能不确定。转换结束后先读取转换结果，再对ADM、ADS和ADPC赋值。如果没有按照上述时序，可能会读取到不正确的转换结果。 | p. 285 □ |
| | | | | 如果从ADCRH读取数据，会产生一个等待周期。当CPU工作在副系统时钟，且外围硬件时钟处于停止状态时，不要从ADCRH读取数据。需要了解详细信息，可参阅 第三十二章 等待注意事项。 | p. 285 □ |
| | | | ADS:<br>模拟输入通道选择寄存器 | 第3～7位必须清零。 | p. 286 □ |
| | | | | 如果对ADS赋值，会产生一个等待周期。当CPU工作在副系统时钟，且外围硬件时钟处于停止状态时，不要对ADS赋值。需要了解详细信息，可参阅 第三十二章 等待注意事项 。 | p. 286 □ |
| | | | ADS:<br>模拟输入通道选择寄存器,<br>ADPC:<br>A/D 端口配置寄存器(ADPC) | 用端口模式寄存器2(PM2)设置一个用于A/D转换通道的端口为输入模式。 | pp. 286, 287 □ |
| | | | ADPC:<br>A/D 端口配置寄存器(ADPC) | 如果将数据写入ADPC，会产生一个等待周期。当CPU工作在副系统时钟，且外围硬件时钟处于停止状态时，不要将数据写入ADPC。需要了解详细信息，可参阅 第三十二章 等待注意事项 。 | p. 287 □ |
| | | | A/D转换器的基本操作 | 必须确保 <1> ～ <5> 的操作时间至少为1 μs 。 | p. 289 □ |
| | | | A/D 转换操作 | 必须确保步骤 <1> ～ <3>的操作时间至少为1 μs。 | p. 293 □ |
| | | | | <1> 可以在<2> 和 <4>之间进行. | p. 293 □ |
| | | | | <1>可以被省略。但在这种情况中（<5>之后）忽略第一个转换结果。 | p. 293 □ |
| | | | | 步骤<6> ～ <9>所经历的时间与使用ADM的第5～1位(FR2 ～ FR0，LV1，LV0)设置的转换时间不同。步骤<8> ～ <9>所经历的时间为FR2 ～ FR0，LV1，LV0设置的转换时间。 | p. 293 □ |
| | | | 在STOP模式下的操作电流 | 在STOP模式中A/D转换器停止操作。此时，将A/D转换器模式寄存器(ADM)的第7位(ADCS)和第0位(ADCE)清零，可以降低操作电流。<br>若要从待机状态重新启动，将中断请求标志寄存器1L (IF1L)的第0位(ADIF)清零(0)，然后开始操作。 | p. 296 □ |
| | 硬件 | | ANI0 ～ ANI7的输入范围 | 观察ANI0 ～ ANI7输入电压的额定范围。如果输入到模拟输入通道的电压大于等于$AV_{REF}$，或者小于等于$AV_{SS}$(即使在绝对最大额定范围之内)，则该通道的转换值不确定。此外，其它通道的转换值也可能会受影响。 | p. 296 □ |
| | 软件 | | 冲突操作 | 转换结束后，通过指令对A/D转换结果寄存器(ADCR，ADCRH)的写操作和ADCR或ADCRH的写操作之间的冲突。<br>ADCR或ADCRH读操作的优先级高。在执行读操作后，才将新的转换结果写入ADCR或ADCRH。 | p. 296 □ |
| | | | | 转换结束后，ADCR或ADCRH的写操作、A/D转换器模式寄存器(ADM)的写操作以及模拟输入通道选择寄存器(ADS)或A/D端口配置寄存器(ADPC)的写操作之间的冲突。<br>ADM、ADS或ADPC的写操作的优先级高。不执行ADCR或ADCRH的写操作，也不会产生转换结束中断信号(INTAD)。 | p. 296 □ |

(12/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 | |
|---|---|---|---|---|---|---|
| 第十二章 | 硬件 | A/D转换器 | 解决噪音问题的方法 | 为了保持10位分辨率，必须注意输入到$AV_{REF}$引脚和ANI0～ANI7引脚的噪音。<br><1> 连接一个低等效电阻的电容和一个优质频率响应到电源上<br><2> 模拟输入源的输出阻抗越大，干扰就越大。为了降低噪音，建议按图13-20所示连接外部C。<br><3> 在转换过程中不要切换引脚。<br><4> 如果在转换开始后立即设置HALT模式，则可以改善精度。 | p. 296 | □ |
| | | | ANI0/P20 ～ ANI7/P27 | 模拟输入引脚(ANI0 ～ ANI7)也可用作输入端口引脚(P20 ～ P27)。<br>当选择ANI0 ～ ANI7中的任意一个通道执行A/D转换时，转换过程中不要访问P20 ～ P27；否则转换分辨率可能会降低。建议从ANI0/P20(离$AV_{REF}$最远)开始，选择引脚用作P20 ～ P27。 | p. 297 | □ |
| | | | | 如果正在进行A/D转换的引脚的相邻引脚有数字脉冲，则由于噪音耦合，有可能得不到预期的A/D转换值。因此在进行A/D转换时不要在相邻引脚引用脉冲。 | p. 297 | □ |
| | | | ANI0 ～ ANI7 引脚的输入阻抗 | 采样期间A/D转换器对采样电容充电，以便进行采样。<br>因此当不进行采样时仅有漏电流经过，而在采样期间则有为电容充电的电流，因此根据是否进行采样，输入阻抗会波动，且无法解决。<br>为了使采样有效，建议模拟输入源的输出阻抗小于等于10 kΩ，并将一个100 pF左右 的电容连接到ANI0 ～ ANI7引脚上(参见 **图12-20**)。 | p. 297 | □ |
| | | | $AV_{REF}$引脚的输入阻抗 | 在$AV_{REF}$与 $AV_{SS}$引脚之间连接几十千欧的串联电阻串。<br>因此，如果参考电压源的输出阻抗很高，当它串联连接到$AV_{REF}$与 $AV_{SS}$引脚之间的串联电阻串时，会导致较大的参考电压误差。 | p. 297 | □ |
| | 软件 | | 中断请求标志(ADIF) | 即使模拟输入通道选择寄存器(ADS)的值被修改，中断请求标志(ADIF)也不会被清零。<br>因此，如果在A/D转换期间有一个模拟输入引脚发生变化，则在ADS被修改之前，模拟输入通道的A/D转换结果和改变前的模拟通道的ADIF可能被设置。此时需要注意是，当ADS修改后立即读取ADIF时，即使修改后的模拟输入的A/D转换尚未结束，也会设置ADIF。<br>当A/D转换停止后又重新开始时，在启动前先对ADIF清零。 | p. 298 | □ |
| | | | A/D转换刚开始时的转换结果 | 在A/D转换开始后，若在ADCE置1后的1 μs 内对ADCS置1，或者ADCE=0时ADCS置1，那么第一次的A/D转换值可能不在额定范围内。可采取措施，如悬挂A/D转换结束中断请求(INTAD)，并删除第1次转换结果。 | p. 298 | □ |
| | | | A/D转换结果寄存器(ADCR, ADCRH)的读取操作 | 当对A/D转换器模式寄存器(ADM)、模拟输入通道选择寄存器(ADS)和A/D端口配置寄存器(ADPC)进行写操作时，ADCR和ADCRH的内容可能不确定。在对ADM、ADS和ADPC进行写操作前且转换结束后读取转换结果。如果采用与上述不同的时序操作可能会读取到不正确的转换结果。 | p. 298 | □ |

(13/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第十三章 | 软件 | 串行接口UART0 | UART 模式 | 如果串行接口UART0使用的时钟未停止(例如处于HALT模式下)，则可继续正常操作。如果时钟已停止(例如处于STOP模式下)，则每个寄存器都停止操作，并且在时钟停止之前及时保存数据。TxD0引脚在时钟停止之前立即保存数据并输出。但在时钟恢复使用后的操作不能得到保证。因此必须复位电路，使POWER0 = 0、RXE0 = 0以及TXE0 = 0。 | p. 300 □ |
| | | | | 先设置POWER0 = 1，然后设置TXE0 = 1(发送)或RXE0 = 1(接收)开始通信。 | p. 300 □ |
| | | | | 由BRGC0设置基本时钟($f_{XCLK0}$)，可使TXE0与RXE0同步。为了能够再次发送或接收，在TXE0或RXE0被清零后，至少要经历两个基本时钟再将TXE0或RXE0置1。如果在不到两个基本时钟时设置TXE0或RXE0，则发送或接收电路可能不被初始化。 | p. 300 □ |
| | | | | 设置TXE0=1之后经过至少一个基本时钟才能发送数据到TXS0。 | pp. 300 303 □ |
| | | | TXS0:<br>发送移位寄存器0 | 设置TXE0=1之后经过至少一个基本时钟($f_{XCLK0}$)才能发送数据到TXS0。<br>在发送完成中断信号(INTST0)产生之前，不要把下一个发送数据写入TXS0。 | p. 303 □ |
| | | | ASIM0:<br>异步串行接口操作模式寄存器0 | 启动发送时，对POWER0置1，然后对TXE0置1。停止发送时，先将TXE0清零，然后将POWER0清零。 | p. 305 □ |
| | | | | 启动接收时，对POWER0置1，然后对RXE0置1。停止接收时，先将RXE0清零，然后将POWER0清零。 | p. 305 □ |
| | | | | 当RxD0引脚输入为高电平时，先对POWER0置1，再对RXE0置1。当输入为低电平，POWER0置1，RXE0置1时，启动接收操作。 | p. 305 □ |
| | | | | 通过BRGC0设置的基本时钟($f_{XCLK0}$)同步TXE0与RXE0。为了能够再次发送或接收，在TXE0或RXE0被清零后，至少要经历两个基本时钟再将TXE0或RXE0置1。如果在不到两个基本时钟的时间内设置TXE0或RXE0，则发送或接收电路可能不被初始化。 | p. 305 □ |
| | | | | 设置TXE0=1之后经过至少一个基本时钟($f_{XCLK0}$)才能设置发送数据到TXS0。 | p. 305 □ |
| | | | | 在对PS01、PS00和CL0位重写之前，先将TXE0和RXE0位清零。 | p. 305 □ |
| | | | | 在重写SL0位时，必须确保TXE0 = 0。总是以“停止位的个数=1”的格式执行接收操作，因此不会受到SL0设置值的影响。 | p. 305 □ |
| | | | | 第0位必须设置为1。 | p. 305 □ |
| | | | ASIS0:<br>异步串行接口接收错误状态寄存器0 | 根据异步串行接口操作模式寄存器0(ASIM0)的PS01和PS00位的内容的不同，对PE0位的操作也有所不同。 | p. 306 □ |
| | | | | 无论停止位的个数是多少，只能将接收数据的第1位作为停止位进行检测。 | p. 306 □ |
| | | | | 如果出现溢出错误，则下一个接收数据不写入接收缓冲寄存器0(RXB0)中，而是被忽略。 | p. 306 □ |
| | | | | 如果从ASIS0读取数据，会产生一个等待周期。当CPU采用副系统时钟操作而停止外围硬件时钟时，不要从ASIS0读取数据。需要了解详细信息，可参见 第三十二章 等待注意事项 。 | p. 306 □ |
| | | | BRGC0:<br>波特率发生器控制寄存器0 | 在重写MDL04 ～ MDL00位时，必须确保ASIM0的第6位(TXE0)和第5位(RXE0) = 0。 | p. 308 □ |
| | 硬件 | | | 波特率为5位计数器输出时钟的1/2。 | p. 308 □ |
| | 软件 | | POWER0, TXE0, RXE0:<br>ASIM0的第7, 6, 5位 | TXE0和RXE0清零后再对POWER0清零，可设置操作停止模式。<br>如果要启动发送或接收，设置POWER0=1，然后设置TXE0或RXE0=1。 | p. 309 □ |
| | | | UART 模式 | 在设置端口模式寄存器和端口寄存器时，要考虑与通信另一方的关系。 | p. 310 □ |

(14/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
| --- | --- | --- | --- | --- | --- |
| 第十三章 | 软件 | 串行接口 UART0 | UART发送 | 在将待发送数据写入TXS0后，在发送完成中断信号(INTST0)产生之前，不要将下一个发送数据写入TXS0。 | p. 313 □ |
| | | | UART接收 | 如果出现接收错误，必须先读取异步串行接口接收错误状态寄存器0 (ASIS0)再读取接收缓冲寄存器0 (RXB0)的内容，才能对出错标志清零。否则，当接收到下一个数据时会产生溢出错误，而且接收错误状态保持不变。 | p. 314 □ |
| | | | UART 接收 | 接收始终按 “停止位的个数 = 1” 的情况执行，第2个停止位被忽略。 | p. 314 □ |
| | | | 波特率误差 | 在发送过程中必须保持波特率误差在接收方允许的误差范围内。 | p. 317 □ |
| | | | | 接收期间的波特率误差必须满足“(4)接收期间允许的波特率范围”中所描述的范围。 | p. 317 □ |
| | | | 接收期间允许的波特率范围 | 必须确保接收期间波特率误差在允许的误差范围内，可用以下公式计算。 | p. 319 □ |
| 第十四章 | 软件 | 串行接口 UART6 | UART模式 | TxD6输出反向操作仅在发送端进行而不在接收端进行。为了能够使用该功能，接收端应准备好接收已反向的数据。 | p. 321 □ |
| | | | | 如果串行接口UART6使用的时钟未被停止(例如，处于HALT模式下)，则继续正常操作。如果串行接口UART6使用的时钟已被停止(例如，处于STOP模式下)，则寄存器停止操作，并且在时钟停止之前及时保存数据。TXD6引脚也在时钟停止之前立即保存数据并输出。但在时钟恢复使用后正常操作不能得到保证。因此必须复位电路，使POWER6 = 0、RXE6 = 0、TXE6 = 0。 | p. 321 □ |
| | | | | 设置POWER6 = 1然后设置TXE6 = 1 (发送)或RXE6 = 1 (接收) ，以启动通信。 | p. 321 □ |
| | | | | 通过基本时钟($f_{XCLK6}$)(由CKSR6设置)对TXE6与RXE6进行同步。若要再次允许发送或接收，必须在TXE6或RXE6清零(0)后至少两个基本时钟时将TXE6或RXE6设置为1。如果在两个基本时钟内设置TXE6或RXE6，则发送电路或接收电路可能不被初始化。 | p. 321 □ |
| | | | | 在设置TXE6 = 1后至少经历一个基本时钟($f_{XCLK6}$)再为TXB6设置发送数据。 | p. 321 □ |
| | | | | 如果连续发送数据，从停止位到下一个起始位的通信时序将延长两个宏操作时钟。但这不会影响到通信结果，因为接收端会在检测到起始位时对时序初始化。如果UART6用于LIN模式，则不能使用连续发送功能。 | p. 321 □ |
| | | | TXB6:<br>发送缓冲寄存器6 | 当异步串行接口发送状态寄存器6(ASIF6)的第1位(TXBF6)=1时不要对TXB6进行写操作。 | p. 327 □ |
| | | | | 在通信过程中(当异步串行接口操作模式寄存器6(ASIM6)的第7位(POWER6)和第6位(TXE6)均为1，或ASIM6的第7位(POWER6)和第5位(RXE6)均为1)，不要通过软件刷新(写入相同值)TXB6。 | p. 327 □ |
| | | | | 在设置TXE6 = 1后，至少经过一个基本时钟($f_{XCLK6}$)，将发送数据写入TXB6。 | p. 327 □ |
| | | | ASIM6:<br>异步串行接口操作模式寄存器6 | 启动发送操作时，对POWER6置1，然后对TXE6置1。停止发送操作时，先将TXE6清零，然后将POWER6清零。 | p. 329 □ |
| | | | | 启动接收操作时，对POWER6置1，然后对RXE6置1。停止接收操作时，先将RXE6清零，然后将POWER6清零。 | p. 329 □ |
| | | | | 当RxD6引脚输入为高电平时，先对POWER6置1，再对RXE6置1。当输入为低电平POWER6置1，RXE6置1时，启动接收操作。 | p. 329 □ |

(15/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第十四章 | 软件 | 串行接口UART6 | ASIM6:<br>异步串行接口操作模式寄存器6 | 通过基本时钟(fxCLK6)(由CKSR6设置)对TXE6与RXE6进行同步。 若要再次允许发送或接收，必须在TXE6或RXE6清零(0)后至少两个基本时钟时将TXE6或RXE6设置为1。如果在两个基本时钟内设置TXE6或RXE6，则发送电路或接收电路可能不被初始化。 | p. 329 ☐ |
| | | | | 设置TXE6 = 1后，至少经过一个基本时钟(fxCLK6)，将发送数据写入TXB6。 | p. 329 ☐ |
| | | | | 在对PS61、PS60和CL6位重写之前，先将TXE6和RXE6位清零。 | p. 329 ☐ |
| | | | | 工作在LIN模式时，PS61与PS60恒为0。 | p. 329 ☐ |
| | | | | 在重写SL6位时，必须确保TXE6 = 0。总是以“停止位的个数=1”的格式执行接收操作，因此接收不会受到SL6设置值的影响。 | p. 329 ☐ |
| | | | | 当重写ISRM6时，RXE6必须为0。 | p. 329 ☐ |
| | | | ASIS6:<br>异步串行接口接收错误状态寄存器6 | 根据异步串行接口操作模式寄存器6(ASIM6)的PS61和PS60位的内容的不同，对PE6位的操作也有所不同。 | p. 330 ☐ |
| | | | | 无论停止位的个数是多少，只将接收数据的第1位作为停止位进行检测。 | p. 330 ☐ |
| | | | | 如果出现溢出错误，则下一个接收数据不写入接收缓冲寄存器6(RXB6)中而是被忽略。 | p. 330 ☐ |
| | | | | 如果从ASIS6读取数据，会产生一个等待周期。当CPU采用副系统时钟而停止外部硬件时钟操作时，不要从ASIS6读取数据。需要了解详细信息，可参见 第三十二章 等待注意事项。 | p. 330 ☐ |
| | | | ASIF6:<br>异步串行接口发送状态寄存器6 | 为了能够连续发送数据，应将第一个发送的数据(第一个字节)写入TXB6中。之后，检查TXBF6标志位，必须为“0”。如果正确，将下一个发送数据(第二个字节)写入TXB6中。如果TXBF6标志为“1”时将数据写入TXB6，则无法保证发送数据的正确性。 | p. 331 ☐ |
| | | | | 如果要在连续发送结束时初始化发送单元，则在产生发送完成中断后必须检查TXSF6标志位，应为“0”。然后执行初始化操作。如果TXSF6标志为“1”时执行初始化操作，则无法保证发送数据的正确性。 | p. 331 ☐ |
| | | | CKSR6:<br>时钟选择寄存器6 | 在重写TPS63 ～ TPS60位时，必须确保POWER6 = 0。 | p. 332 ☐ |
| | | | BRGC6:<br>波特率发生器控制寄存器6 | 在重写MDL67 ～ MDL60位时，必须确保ASIM0的第6位(TXE6)和第5位(RXE6) = 0。 | p. 333 ☐ |
| | 硬件 | | | 波特率为8位计数器输出时钟的1/2。 | p. 333 ☐ |
| | 软件 | | ASICL6:<br>异步串行接口控制寄存器6 | 通信过程中(当异步串行接口操作模式寄存器6(ASIM6)的第7位(POWER6)和第6位(TXE6)均为1，或ASIM6的第7位(POWER6)和第5位(RXE6)均为1)，可由软件刷新(写入相同值)ASICL6。但是，由于可能会重新触发SBF接收和SBF发送，因此在SBF接收期间(SBRT6 = 1)或SBF发送期间通过刷新操作不要将SBRT6与SBTT6都设置为1(直到由于SBTT6=1而产生INTST6为止)。 | p. 334 ☐ |
| | | | | 如果出现SBF接收错误，则返回到SBF接收模式，并保持SBRF6标志位状态(1)不变。 | p. 335 ☐ |
| | | | | 在设置SBRT6之前，必须确保ASIM6的第7位(POWER6)和第5位(RXE6)为1。在设置SBRT6=1后SBF接收完成(产生中断请求信号之前)之前不要将SBRT6清零。 | p. 335 ☐ |
| | | | | SBRT6的读取值恒为0。在SBF正确接收后SBRT6自动清零。 | p. 335 ☐ |
| | | | | 在设置SBTT6=1之前，必须确保ASIM6的第7位(POWER6)和第6位(TXE6)为1。在设置SBTT6=1之后SBF接收完成(产生中断请求信号之前)之前不要将SBTT6清零。 | p. 335 ☐ |
| | | | | SBTT6的读取值恒为0。并在SBF发送结束后自动清零。 | p. 335 ☐ |

(16/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第十四章 | 软件 | 串行接口UART6 | ASICL6:<br>异步串行接口控制寄存器6 | 在接收期间不要设置SBRT6=1，而在发送期间也不要设置SBTT6=1。 | p. 335 □ |
| | | | | 在重写DIR6和TXDLV6之前，应对TXE6 和RXE6清零。 | p. 335 □ |
| | | | ASIM6的第7、6和5位(POWER6、TXE6和RXE6) | TXE6和RXE6清零后再对POWER6清零，可设置操作停止模式。<br>如果要启动发送或接收操作，则先设置POWER6=1，然后设置TXE6=1或RXE6=1。 | p. 337 □ |
| | | | UART 模式 | 在设置端口模式寄存器和端口寄存器时，要考虑与通信另一方的关系。 | p. 338 □ |
| | | | 校验方式与操作 | 在LIN模式下，PS61与PS60恒为0。 | p. 341 □ |
| | | | 连续发送 | 在连续发送期间，ASIF6寄存器的TXBF6和TXSF6标志从“10” 变成 “11”， 再变成“01”。因此检查状态时，不能通过比较TXBF6和TXST6来判断。在连续发送期间只读TXBF6标志。 | p. 343 □ |
| | | | | 在LIN模式下，不能使用连续发送功能。在将发送数据写入发送缓冲寄存器6(TXB6)之前，异步串行接口发送状态寄存器6(ASIF6)的值必须为00H。 | p. 343 □ |
| | | | | 为了能够连续发送数据，应将第一个发送的数据(第一个字节)写入TXB6中。之后，检查TXBF6标志位，必须为“0”。如果正确，将下一个发送数据(第二个字节)写入TXB6中。如果TXBF6标志为“1”时将数据写入TXB6，则无法保证发送数据的正确性。 | p. 343 □ |
| | | | | 如果要在连续发送结束时初始化发送单元，则在产生发送完成中断后必须检查TXSF6标志位，应为“0”。然后执行初始化操作。如果TXSF6标志为“1”时执行初始化操作，则无法保证发送数据的正确性。 | p. 343 □ |
| | | | | 在连续发送期间，当发送完一帧数据后执行INTST6中断服务之前下一个发送操作可能已经完成。计数测量时，可通过开发一个可以算发送数据的个数的程序或者通过参考TXSF6标志来实现检测。 | p. 343 □ |
| | | | 正常接收 | 如果出现接收错误，则先后读取ASIS6和RXB6，可将出错标志清零。否则，当接收到下一个数据时会产生溢出错误，而且接收错误状态保持不变。 | p. 347 □ |
| | | | | 接收始终按“停止位的个数 = 1”的情况执行，第2个停止位被忽略。 | p. 347 □ |
| | | | | 在读取RXB6之前，必须先读取异步串行接口接收错误状态寄存器6(ASIS6)。 | p. 347 □ |
| | | | 波特率误差 | 发送期间必须保持波特率误差在接收端允许的误差范围内。 | p. 353 □ |
| | | | | 接收期间，波特率误差必须满足“(4)接收期间允许的波特率范围”中所描述的范围。 | p. 353 □ |
| | | | 接收期间允许的波特率范围 | 必须确保接收期间波特率误差在允许的误差范围内，可用以下公式计算。 | p. 355 □ |
| 第十五章 | 软件 | 串行接口CSI10 | SOTB10:<br>发送缓冲寄存器10 | 当CSOT10 = 1（在串行通信期间）时不要访问SOTB1n。 | p. 360 □ |
| | | | SIO10:<br>串行I/O移位寄存器10 | 当CSOT10 = 1（在串行通信期间）时不要访问SIO10。 | p. 360 □ |
| | | | CSIM10:<br>串行操作模式寄存器10 | 第5位必须清零。 | p. 361 □ |

(17/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第十五章 | 软件 | 串行接口 CSI10 | CSIC10: 串行时钟选择寄存器10 | 当CSIE10=1时（允许操作），不要对CSIC10进行写操作。 | p. 362 □ |
| | | | | 若将P10/SCK10/TxD0和P12/SO10用作通用端口，设置CSIC10为默认状态(00H)。 | p. 362 □ |
| | | | | 复位后数据时钟的相位类型为类型1。 | p. 362 □ |
| | | | 3线串行I/O模式 | 在设置端口模式寄存器和端口寄存器时，要考虑与通信另一方的关系。 | p. 365 □ |
| | | | 通信操作 | 当CSOT10 = 1时（串行通信期间），不要访问控制寄存器和数据寄存器。 | p. 366 □ |
| | | | SO10输出 | 如果对TRMD10、DAP10和DIR10进行写操作，则SO10的输出值将发生改变。 | p. 374 □ |
| 第十六章 | 软件 | 串行接口 IIC0 | IIC0: IIC移位寄存器0 | 在数据传送期间不要对IIC0进行写操作。 | p. 378 □ |
| | | | | 仅在等待期间才能对IIC0进行读写操作。除等待期间外，禁止在其他的通信状态下访问IIC0。然而，作为主设备时，在通信触发位(STT0)被设为1之后IIC0只能被写入一次。 | p. 378 □ |
| | | | IICC0: IIC 控制寄存器0 | 允许$I^2C$ 操作(IICE0 = 1)后，在SCL0为高电平并且SDA0为低电平时，立即检测到起始条件。允许$I^2C$操作(IICE0 = 1)后，立即通过1位存储器操作指令设置LREL0(1)。 | p. 382 □ |
| | | | | 在第9个时钟，当IIC状态寄存器0(IICS0)的第3位(TRC0)被设为1和WREL0设置为1时，，取消等待状态后，TRC0被清零，SDA0线被设为高阻抗状态。 | p. 385 □ |
| | | | IICS0: IIC状态寄存器0 | 如果从IICS0中读取数据，将产生一个等待周期。当CPU使用副系统时钟且外围硬件时钟停止时，不要从IICS0中读取数据。详见　第三十二章 等待注意事项　。 | p. 386 □ |
| | | | IICF0: IIC标志寄存器0 | 仅当操作停止(IICE0 = 0)时才可对STCEN进行写操作。 | p. 389 □ |
| | | | | 当STCEN = 1时不论总线处于何种状态，总线释放状态(IICBSY = 0)都可以识别，因此当第一个起始条件产生(STT0 = 1)时，必须确保没有第三方通信，以防通信被破坏。 | p. 389 □ |
| | | | | 仅当操作停止(IICE0 = 0)时才可对IICRSV进行写操作。 | p. 389 □ |
| | | | 选择时钟设置 | 允许操作(通过设置IIC控制寄存器0 (IICC0)的第7位(IICE0)为1实现)前，使用CLX0、SMC0、CL01和CL00决定$I^2C$ 的传送时钟频率。若改变传送时钟频率，必须把IICE0清零。 | p. 392 □ |
| | | | 当STCEN (IIC标志寄存器0 (IICF0)的第1位) = 0时 | 在允许$I^2C$操作(IICE0 = 1)后，无论总线的实际状态如何都可以识别总线通信状态(IICBSY (IICF0的第6位) = 1)。从检测到非停止条件模式变为主设备通信模式时，首先产生一个停止条件以释放总线，然后进行主设备通信。<br>当使用多个主设备时，如果总线未被释放(没有检测到停止条件)，则不能进行主设备通信。<br>按下列步骤产生停止条件。<br><1>　设置IIC时钟选择寄存器0 (IICCL0)。<br><2>　设置 IIC控制寄存器0 (IICC0) 的第7位(IICE0)为 1。<br><3>　设置 IICC0 的第0位(SPT0)为 1。 | p. 409 □ |
| | | | 当STCEN = 1 | 在允许$I^2C$操作(IICE0 = 1)后，无论总线的实际状态如何都可以识别总线通信状态(IICBSY (IICF0的第6位) = 1)。要产生第一个起始条件 (STT0 (IIC控制寄存器0 (IICC0)的第1位) = 1), 必须确认总线已被释放，以免干扰其他通信。 | p. 409 □ |

(18/25)

<table>
<tr><th>章节</th><th>分类</th><th>功能</th><th>功能描述</th><th>注意事项</th><th>页码</th></tr>
<tr><td rowspan="4">第十六章</td><td rowspan="4">软件</td><td rowspan="4">串行接口IIC0</td><td>如果其他I²C 通信已经进行</td><td>如果SDA0引脚为低电平同时SCL0引脚为高电平时允许I²C 操作，且设备已进行通信，则I²C 的宏可以识别SDA0引脚已经变为低电平(检测一个起始条件)。如果此时总线上的值被识别为一个扩展码，则返回ACK信号，但这会干扰其他I²C 通信。为了避免这种情况，可按下列步骤启动I²C。<br><1> 当检测到停止条件时，将IICC0的第4位(SPIE0)清零，禁止产生中断请求信号(INTIIC0)。<br><2> 设置IICC0的第7位(IICE0)为1，允许I²C的操作。<br><3> 等待对启动条件的检测。<br><4> ACK信号返回之前(设置IICE0为1之后的4 到80 个时钟)设置IICC的第6位(LREL0) 为1，强行禁止检测。</td><td>p. 409 □</td></tr>
<tr><td>传输时钟频率设置</td><td>允许操作之前(IICE0 = 1)通过使用 SMC0, CL01, CL00 (IICL0的第3，1和0位)和CLX0 (IICX0的第0位)决定传送时钟频率。要改变传送时钟频率，必须清零IICE0。</td><td>p. 409 □</td></tr>
<tr><td>STT0, SPT0:<br>IIC控制寄存器0(IICC0)的第1, 2位</td><td>在对STT0和SPT0 (IICC0的第1和第0位)设置之后和清零之前，禁止再次设置。</td><td>p. 410 □</td></tr>
<tr><td>发送接收</td><td>当预约发送时，设置SPIE0 (IICL0的第4位)为1，以允许在检测到停止条件时产生中断请求。中断请求产生后，将通信数据写入IIC0，传送过程开始。如果检测到停止条件时不产生中断请求，设备将处于等待状态，因为开始通信时不产生中断请求。但是，由软件检测MSTS0 (IICS0的第7位)时不必设置SPIE0为1。</td><td>p. 410 □</td></tr>
<tr><td rowspan="10">第十七章</td><td rowspan="10">软件</td><td rowspan="10">乘法器/除法器( 仅μPD78F0514, 78F0515, 和78F0515D)</td><td rowspan="2">SDR0:<br>余数寄存器0</td><td>在操作期间(乘法器/除法器控制寄存器0 (DMUC0)的第7位(DMUE)= 1)从SDR0中读取的值不确定。</td><td>p. 448 □</td></tr>
<tr><td>当启动操作时(当DMUE=1时)，SDR0被复位。</td><td>p. 448 □</td></tr>
<tr><td rowspan="3">MDA0H, MDA0L:<br>乘/除数寄存器A0</td><td>在乘法模式中(乘法器/除法器控制寄存器0 (DMUC0) = 81H)启动操作时，MDA0H被清零。</td><td>p. 448 □</td></tr>
<tr><td>在操作期间(乘法器/除法器控制寄存器0 (DMUC0)的第7位(DMUE)= 1)不要修改MDA0的值。此时即使执行修改操作，结果也不确定。</td><td>p. 448 □</td></tr>
<tr><td>在操作期间(当DMUE = 1)从MDA0读取的值不确定。</td><td>p. 448 □</td></tr>
<tr><td rowspan="2">MDB0:<br>乘/除数寄存器B0</td><td>在操作期间(乘法器/除法器控制寄存器0 (DMUC0)的第7位(DMUE)= 1)不要修改MDB0的值。此时即使执行修改操作，结果也不确定。</td><td>p. 449 □</td></tr>
<tr><td>在除法模式中不要对MDB0清零(0000H)。如果清零，则存储在MDA0 与 SDR0中的操作结果将不确定。</td><td>p. 449 □</td></tr>
<tr><td rowspan="3">DMUC0:<br>乘法器/除法器控制寄存器0</td><td>如果操作期间(当DMUE = 1)DMUE被清零，则操作结果不确定。如果设置了中断标志，在执行清零指令时操作完成，操作结果可以得到保证。</td><td>p. 450 □</td></tr>
<tr><td>在操作期间(当DMUE = 1)时，不要修改DMUSEL0的值。如果修改了，则存放在乘/除数寄存器A0 (MDA0)和余数寄存器0 (SDR0)中的操作结果不确定。</td><td>p. 450 □</td></tr>
<tr><td>如果在操作期间(当DMUE = 1)将DMUE清零，则停止操作。若要再次执行操作，应对乘/除数寄存器A0 (MDA0)、乘/除数寄存器B0 (MDB0)和乘法器/除法器控制寄存器0 (DMUC0)进行设置，然后启动操作(通过设置DMUE = 1)。</td><td>p. 450 □</td></tr>
</table>

(19/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第十八章 | 软件 | 中断功能 | 1F0L, 1F0L, 1F1L, 1F1H:<br>中断请求标志寄存器 | 对44引脚产品，必须对IF1L的第6和7位清零。<br>对48引脚产品，必须对IF1L的第7位清零。 | p. 460 □ |
| | | | | 必须对1F1L第7 位和 IF1H的第1 ~ 7 位清零。 | p. 460 □ |
| | | | | 当退出待机模式要使用定时器、串行接口或A/D转换器时，在将中断请求标志清零后对这些部件操作一次。噪音可能会干扰中断请求标志。 | p. 460 □ |
| | | | | 修改中断请求标志寄存器中的标志时, 使用一位存储器操作指令 (CLR1) 。当用C语言描述时,由于编译后的汇编指令必须是一位存储器操作指令(CLR1),所以应该使用一位操作指令，如“IF0L.0 = 0;” 或 “_asm(“clr1 IF0L, 0”);”。<br>如果一条8位存储器操作指令如“IF0L &= 0xfe;”,用C语言描述,则编译后将被转换为三条汇编指令：<br>mov a, IF0L<br>and a, #0FEH<br>mov IF0L, a<br>在这个例子中，当处于“mov a, IF0L”和 “mov IF0L, a”之间的时序时，即使同一中断请求标志寄存器(IF0L)的另一位请求标志被设置为1，则该请求标志也将被“mov IF0L, a”清零。因此在C语言中使用8位存储器操作指令时必须小心。 | p. 461 □ |
| | | | MK0L, MK0H, MK1L, MK1H:<br>中断屏蔽标志寄存器 | 对44引脚产品，必须将MK1L的第6和7位置1。<br>对48引脚产品，必须将MK1L的第7位置1。 | p. 462 □ |
| | | | | 必须对MK1H的第1位置1。 | p. 462 □ |
| | | | PR0L, PR0H, PR1L, PR1H:<br>优先级指定标志寄存器 | 对44引脚产品，必须将PR1L的第6和7位置1。<br>对48引脚产品，必须将PR1L的第7位置1。 | p. 463 □ |
| | | | | 必须对PR1H的第1位置1。 | p. 463 □ |
| | | | EGP, EGN:<br>外部中断上升沿允许寄存器 、 外部中断下降沿允许寄存器 | 对44引脚产品，必须对EGP 和 EGN的第6和7位清零。<br>对48引脚产品，必须对EGP 和 EGN的第7位清零。 | p. 464 □ |
| | | | | 在从外部中断功能切换到端口功能时可能会检测到脉冲沿，因此通过将EGPn 和 EGNn 清零可选择端口模式。 | p. 464 □ |
| | | | 软件中断请求响应 | 不能使用RETI指令从软件中断返回。 | p. 468 □ |
| | | | 保持中断请求 | BRK指令不属于上述列出的中断请求保持指令。但通过执行BRK指令激活的软件中断会将IE标志清零。因此，即使在执行BRK指令期间产生可屏蔽中断请求，该中断请求也不会被响应。 | p. 472 □ |
| 第十九章 | 软件 | 按键中断功能 | KRM:<br>按键返回模式寄存器 | 如果KRM0 ～ KRM3中任何一位被设置为1，则将上拉电阻寄存器7(PU7)的0 ～ 3位(PU70 ～ PU73)的相应位设置为1。 | p. 474 □ |
| | | | | 如果KRM发生变化，则中断请求标志可能被设置。因此可以先禁止中断，再修改KRM。先将中断请求标志清零，再允许中断。 | p. 474 □ |
| | | | | 在按键中断模式中没有被使用的位可用作通用端口。 | p. 474 □ |

(20/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第二十章 | 软件 | 待机功能 | 待机功能 | 仅当CPU使用主系统时钟时，才能使用STOP模式。不能停止副系统时钟的振荡。而当CPU使用主系统时钟或副系统时钟时，均可以使用HALT模式。 | p. 475 ☐ |
| | | | | 当切换到STOP模式时，在执行STOP指令前必须停止使用主系统时钟的外围硬件的操作。 | p. 475 ☐ |
| | | | | 当使用待机功能时，建议采用以下步骤降低A/D转换器的操作电流：首先将A/D转换器模式寄存器（ADM）的第7位（ADCS）和第0位（ADCE）清零以停止A/D转换操作，然后执行STOP指令。 | p. 475 ☐ |
| | | | OSTC:<br>振荡稳定时间计数器的状态寄存器 | 在经历上述时间后，从MOST11起各位被设置为1，并保持为1。 | p. 476 ☐ |
| | | | | 振荡稳定时间计数器的计数达到OSTS设置的振荡稳定时间。如果已进入STOP模式并在CPU使用内部高速振荡时钟时释放STOP模式，则可以按以下方式设置振荡稳定时间。<br>• 预期的OSTC振荡稳定时间 ≤ OSTS设置的振荡稳定时间<br>因此需要注意，在释放STOP模式后，只有通过OSTS设置的振荡稳定时间期间的状态被设置到OSTC。 | p. 476 ☐ |
| | 硬件 | | | X1振荡稳定等待时间不包括从释放STOP模式到时钟振荡启动这段时间（即下图“a”所示的部分）。 | p. 476 ☐ |
| | 软件 | | OSTS:<br>振荡稳定时间选择寄存器 | 如果在X1时钟用作CPU时钟时要设置STOP模式，则应在执行STOP指令之前设置OSTS。 | p. 477 ☐ |
| | | | | 在X1时钟振荡稳定期间不要修改OSTS寄存器的值。 | p. 477 ☐ |
| | | | | 振荡稳定时间计数器的计数达到OSTS设置的振荡稳定时间。如果已进入STOP模式，并在CPU使用内部高速振荡时钟时释放STOP模式，则可以按以下方式设置振荡稳定时间。<br>• 预期的OSTC振荡稳定时间 ≤ OSTS设置的振荡稳定时间<br>因此需要注意，在释放STOP模式后，只有通过OSTS设置的振荡稳定时间期间的状态被设置到OSTC。 | p. 477 ☐ |
| | 硬件 | | | X1时钟振荡等待时间不包括从释放STOP模式到时钟振荡启动这段时间（即下图“a”所示的部分）。 | p. 477 ☐ |
| | 软件 | | STOP模式 | 由于中断请求信号用于释放待机模式，因此如果一个中断源的中断请求标志被设置且中断屏蔽标志被清零时，则立即释放待机模式。因此，在执行STOP指令后，立即将STOP模式复位到HALT模式，并且在经历了OSTS设置的等待时间后，系统返回操作模式。 | p. 483 ☐ |
| | | | | 在STOP模式释放后，要使用在STOP模式下停止操作的外围硬件，以及在STOP模式下时钟停止操作的外围硬件，必须重新启动外围硬件。 | p. 485 ☐ |
| | | | | 即使通过可选字节选择“可由软件停止内部低速振荡器”，在STOP模式下，内部低速振荡时钟仍继续振荡，并保持STOP模式设置前的状态。要在STOP模式下停止内部低速振荡器的振荡，应通过软件停止该振荡器，然后执行STOP指令。 | p. 485 ☐ |
| | | | | 当CPU使用高速系统时钟(X1振荡)时，为了在STOP模式释放后缩短振荡稳定时间，可在执行STOP指令前将CPU时钟暂时切换到内部高速振荡时钟。STOP模式释放后，在将CPU时钟从高速振荡时钟切换到高速系统时钟 (X1振荡) 之前，应使用振荡稳定时间计数器的状态寄存器(OSTC)检测振荡稳定时间。 | p. 485 ☐ |

(21/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第二十章 | 软件 | 待机功能 | STOP模式 | 当AMPH = 1时执行STOP指令，当内部高速振荡时钟作为CPU时钟时，在STOP模式释放后CPU时钟停止4.06 ～ 16.12 µs，或者当高速系统时钟（外部时钟输入）作为CPU时钟时，CPU时钟停止160个外部时钟周期的时间。 | p. 485 ☐ |
| 第二十一章 | 硬件 | 复位功能 | – | 对于外部复位，输入RESET引脚的低电平的时间至少应为10 µs。 | p. 489 ☐ |
| | | | | 复位输入期间，X1时钟、XT1时钟、内部高速振荡时钟和内部低速振荡时钟停止振荡。外部主系统时钟输入和外部副系统时钟输入无效。 | p. 489 ☐ |
| | | | | 在通过复位释放STOP模式时，复位输入期间保持STOP模式的内容。但P130（低电平输出）除外的端口引脚变为高阻态。 | p. 489 ☐ |
| | | | 复位功能的框图 | LVI电路内部复位功能不能复位LVI电路。 | p. 490 ☐ |
| | | | 看门狗定时器溢出 | 看门狗定时器内部复位功能也可以复位看门狗定时器。 | p. 491 ☐ |
| | 软件 | | RESF:<br>复位控制标志寄存器 | 不能通过1位存储器操作指令读取数据。 | p. 497 ☐ |
| 第二十二章 | 软件 | 上电清零电路 | – | 如果由POC电路产生内部复位信号，则将复位控制标志寄存器（RESF）清零（00H）。 | p. 498 ☐ |
| | | | | 在释放复位状态后通过软件设置低电压检测电路 (参见第二十三章 低电压检测电路)。 | pp. 500, 501 ☐ |
| | | | In 2.7 V/1.59 V POC 模式 | 在电源电压抵达1.59V (TYP.) 以后，需要一个1.93 到 5.39 ms的电压振荡稳定时间。如果在1.93 ms时间以内，电源电压由1.59 V (TYP.) 上升到 2.7 V (TYP.)， 则在复位处理之前，自动产生一个0 到 5.39 ms的电源振荡稳定时间。 | p. 501 ☐ |
| | | | 上电清零电路使用注意事项 | 在系统中，如果供电电压 ($V_{DD}$) 接近POC检测电压 ($V_{POC}$) 时的一段时间内产生波动，则系统可能会反复复位和释放复位。在这种情况下，可采用下列方法任意设置从复位释放到微控制器的启动所需要的时间。 | p. 502 ☐ |
| 第二十三章 | 软件 | 低电压检测电路 | LVIM:<br>低电压检测寄存器 | 在执行以下任一过程后，停止LVI。<br>• 当使用8位操作指令时: 00H → LVIM。<br>• 当使用1位存储器操作指令时: 对LVION清零。 | p. 506 ☐ |
| | 硬件 | | | 外部输入引脚的输入电压(EXLVI)必须为： EXLVI < $V_{DD}$。 | p. 506 ☐ |
| | 软件 | | LVIS:<br>低电压检测等级选择寄存器 | 第4 ～ 7位必须清零。 | p. 507 ☐ |
| | | | | LVI操作期间不要修改LVIS的值。 | p. 507 ☐ |
| | | | | 当检测到外部输入引脚的输入电压(EXLVI)时，检测电压($V_{EXLVI}$ = 1.21 V (TYP.))恒定。因此，不必设置LVIS。 | p. 507 ☐ |
| | | | 当检测供电电压($V_{DD}$)等级时 | 必须执行过程<1>。当LVIMK = 0时，在过程<4>之后可能立即会产生中断。 | p. 509 ☐ |
| | | | | 当LVIM=1时如果供电电压（$V_{DD}$） ≥ 检测电压（$V_{LVI}$），则不会产生内部复位信号。 | p. 509 ☐ |
| | | | 检测外部输入引脚的输入电压(EXLVI) 等级 | 必须执行<1>。当LVIMK = 0时，在完成<3>后可能立即会产生中断。 | p. 512 ☐ |
| | | | | I如果在LVIMD=1时，外部输入引脚的输入电压(EXLVI) ≥ 检测电压($V_{EXLVI}$ = 1.21 V (TYP.))，则不会产生内部复位信号。 | p. 512 ☐ |
| | 硬件 | | | 必须保证外部输入引脚的输入电压(EXLVI) ：EXLVI < $V_{DD}$ | p. 512 ☐ |

(22/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第二十三章 | 硬件 | 低电压检测电路 | 检测外部输入引脚的输入电压(EXLVI) 等级 | 必须保证外部输入引脚的输入电压(EXLVI) ：EXLVI < $V_{DD}$ | p. 517 □ |
| | 软件 | | 低电压检测电路的注意事项 | 在系统中，如果供电电压（$V_{DD}$）在接近LVI检测电压（$V_{LVI}$）的一段时间内产生波动，这时根据低电压检测电路的使用情况进行如下操作。<br>(1) 用于复位<br>系统可能会反复进行复位和释放复位。<br>在这种情况下，可采用下面的方法（1）任意设置从复位释放到微控制器的启动所经历的时间。<br>(2) 用于中断<br>可能会频繁产生中断请求。可采用方法（2）中的(b)。 | p. 519 □ |
| 第二十四章 | 软件 | 选项字节 | 0082H, 0083H/ 1082H, 1083H | 确保将地址 0082H 和 0083H的内容清零 (0082H/1082H 和 0083H/1083H，当使用启动交换功能时)。 | p. 522 □ |
| | | | 0080H/1080H | 因为地址0080H 和 1080H在启动交换操作中切换，所以要将0080H 的值赋值给1080H。 | p. 522 □ |
| | | | 0081H/1081H | POCMODE只能通过专用Flash编程器写入。在自编程中(此时，设定为1.59V POC模式(缺省))，自编程或者启动交换操作都不能用于设定POCMODE的值。以上，因为在启动交换操作中1081H的值将拷贝到0081H中，所以推荐使用启动交换功能时，先将0081H 的值赋给1081H。 | p. 522 □ |
| | | | 0084H/1084H | 对于没有配备片上调试功能的产品(μPD78F0511, 78F0512, 78F0513, 78F0514, 和78F0515)，必须将00H赋给0084H单元(禁止片上调试操作)。由于在引导交换时0084H和1084H 要相互切换，因此也要对1084H设置00H。 | p. 523 □ |
| | | | | 对于具有片上调试功能的产品(μPD78F0513D, 78F0515D)，如果要使用它的片上调试功能，需要对0084H单元赋值02H或03H。由于在引导交换时0084H和1084H 要相互切换，因此也要对1084H设置与0084H相同的值。 | p. 523 □ |
| | | | 0080H/1080H | 禁止设置：WDCS2 = WDCS1 = WDCS0 = 0且WINDOW1 = WINDOW0 = 0 | p. 524 □ |
| | | | | Flash存储器自编程和EEPROM 仿真期间看门狗定时器继续工作。处理过程中中断响应时间将被延迟。设置溢出时间和窗口大小时应考虑到这一延迟情况。 | p. 524 □ |
| | | | | 如果LSROSC = 0(可由软件停止振荡)，则无论内部振荡模式寄存器(RCM)的第0位(LSRSTOP)为何值，在HALT和STOP模式下对看门狗定时器不提供计数时钟。<br>当8位定时器H1使用内部低速振荡时钟时，即使在HALT/STOP模式下也对8位定时器H1提供计数时钟。 | p. 524 □ |
| | | | | 必须对第7位清零(0)。 | p. 524 □ |
| | | | 0081H/1081H | 必须将第7 ～ 1位清零(0)。 | p. 525 □ |
| 第二十五章 | 软件 | Flash 存储器 | IMS:<br>内部存储器容量切换寄存器,<br>IXS:<br>内部扩展RAM容量切换寄存器 | 复位释放后必须按照表25-1对每个产品进行设置。 | p. 527 □ |
| | | | | 在复位释放后必须如表25-2所示设置每种产品的值。 | p. 528 □ |
| | | | 操作时钟 | 在使用CSI10时，只能使用内部高速振荡时钟 ($f_{RH}$)。 | p. 539 □ |
| | | | | 在使用UART6时，只能使用X1时钟 ($f_X$)或外部主系统时钟 ($f_{EXCLK}$)。 | p. 539 □ |

(23/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第二十五章 | 软件 | Flash 存储器 | X1, P31 引脚的处理 | 对于具有48KB或更大Flash存储器而没有片上调试功能的产品 (μPD78F0514 和 78F0515)，有一个“I”, “K”, 或 “E” 的产品等级，对于具有片上调试功能的产品 (μPD78F0513D 和 78F0515D)，在使用Flash编程器写Flash存储器时，按以下连接 P31/INTP2/OCD1A[注] 和 P121/X1/OCD0A[注]。<br>• P31/INTP2/OCD1A[注]: 通过电阻连接到 Vss (10 kΩ: 推荐)。<br>• P121/X1/OCD0A[注]: 当使用此引脚作为一个端口时，通过电阻连接到 Vss (10 kΩ: 推荐)(在输入模式)或悬空(在输出模式)。<br>当通过自编程方式写入Flash存储器时，以上连接并不需要。 | p. 540 ☐ |
| | | | 选择通信模式 | 当选择UART6时，在接收到FLMD0脉冲后，根据专用Flash 存储器编程器发送的复位命令计算接收时钟。 | p. 542 ☐ |
| | 硬件 | | 安全设置 | 在对一次性擦除进行安全性设置后，不能再对该设置执行擦除操作。此外，由于禁止执行擦除命令，即使执行了写命令，与已经写到Flash 存储器中的数据不同的数据也不能被写入。 | p. 544 ☐ |
| | | | | 如果对重写引导簇0进行了安全性设置，则不能再对该设备的引导簇0进行重写。 | p. 544 ☐ |
| | | | 通过自编程进行Flash 存储器编程 | 当CPU使用副系统时钟时不能使用自编程功能。 | p. 547 ☐ |
| | | | | 自编程期间FLMD0引脚应输入高电平。 | p. 547 ☐ |
| | 软件 | | | 在启动自编程之前必须执行DI指令。<br>自编程功能检查中断请求标志 (IF0L, IF0H, IF1L和IF1H)。如果产生了中断请求，则停止自编程操作。 | p. 547 ☐ |
| | | | | 即使在DI状态下也可以通过未屏蔽的中断请求停止自编程。要避免这种情况，可以通过使用中断屏蔽标志寄存器(MK0L, MK0H, MK1L和MK1H)屏蔽该中断。 | p. 547 ☐ |
| | | | | 将用于自编程的入口程序分配在公共区域(0000H ～ 7FFFH)。 | p. 547 ☐ |
| 第二十六章 | 硬件 | 片上调试功能 (仅μPD78F0513D 和 78F0515D) | μPD78F0513D, 78F0515D | μPD78F0513D 和 78F0515D具有片上调试功能。由于在使用片上调试功能后产品的稳定性不能得到保证，因此在给定flash存储器可重写次数的情况下不要将该产品用于大规模生产。NEC Electronics不接受有关该产品的投诉。 | p. 555 ☐ |
| | | | 当使用 OCD0A/X1 和 OCD0B/X2时 | 在片上调试期间从OCD0A/X1引脚输入时钟。 | p. 555 ☐ |
| | | | | 通过外部下拉OCD1A/P31引脚或通过使用P130 [注2]引脚的外部电路(当该设备复位时输出低电平)，控制OCD0A/X1和OCD0B/X2引脚 。 | p. 555 ☐ |
| 第二十八章 | 硬件 | 电气特性 (标准产品) | μPD78F0513D, 78F0515D | μPD78F0513D 和 78F0515D具有片上调试功能。不要将该产品用于大规模生产，因为在给定flash存储器重写次数的条件下，使用片上调试功能后，产品的可靠性得不到保证。NEC Electronics 不接受有关该产品的任何投诉。 | p. 571 ☐ |
| | | | 最大额定值 | 任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因而，必须保证产品在不超过最大额定值的条件下使用。 | pp. 571, 572 |

(24/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第二十八章 | 硬件 | 电气特性 (标准产品) | X1振荡器特性 | 在使用X1时钟时，上图中被虚线包围的部分的配线应按照如下配线方法配线，以防止连接线电容产生不利影响。<br>• 连接线越短越好。<br>• 连接线不应与其他信号线交叉。<br>• 流经的电流变化较大的信号线不要在振荡器周围布线。<br>• 要保持振荡器电容器的接地点电压与Vss相同。<br>• 不要将电容的地信号接入大电流地。<br>• 不要从振荡器获取信号。 | p. 573 □ |
| | | | | 由于复位后CPU使用内部高速振荡时钟进行操作，因此用户要使用振荡稳定时间计数器的状态寄存器（OSTC）检测X1时钟振荡稳定时间。在充分评估了所使用的振荡器的振荡稳定时间后再确定OSTC和振荡稳定时间选择寄存器（OSTS）的振荡稳定时间。 | p. 573 □ |
| | | | XT1 荡器特性 | 在使用X1时钟时，上图中被虚线包围的部分的配线应按照如下配线方法配线，以防止连接线电容产生不利影响。<br>• 连接线越短越好。<br>• 连接线不应与其他信号线交叉。<br>• 流经的电流变化较大的信号线不要在振荡器周围布线。<br>• 要保持振荡器电容器的接地点电压与Vss相同。<br>• 不要将电容的地信号接入大电流地。<br>• 不要从振荡器获取信号。 | p. 574 □ |
| | | | | XT1振荡器是一个低振幅电路，用于降低功耗，但由于比X1振荡器更容易受到噪音干扰，因此在使用XT1时钟时应特别注意连线方式。 | p. 574 □ |
| | | | 振荡器常数 | 以上是根据振荡器厂家提供的在特定环境下的参考常数。如果在实际应用中需要优化振荡器特性，可以请厂家为应用电路做评价。振荡电压和频率仅仅包括振荡器特性。使用78K0/KC2以便内部操作条件包含在DC和AC特性中。 | pp. 575, 576 □ |
| 第二十九章 | 硬件 | 电气特性 ((A)级产品，目标系统) | 最大额定值 | 任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因而，必须保证产品在不超过最大额定值的条件下使用。 | pp.592, 593 □ |
| | | | X1振荡器特性 | 在使用X1时钟时，上图中被虚线包围的部分的配线应按照如下配线方法配线，以防止连接线电容产生不利影响。<br>• 连接线越短越好。<br>• 连接线不应与其他信号线交叉。<br>• 流经的电流变化较大的信号线不要在振荡器周围布线。<br>• 要保持振荡器电容器的接地点电压与Vss相同。<br>• 不要将电容的地信号接入大电流地。<br>• 不要从振荡器获取信号。 | p.594 □ |

(25/25)

| 章节 | 分类 | 功能 | 功能描述 | 注意事项 | 页码 |
|---|---|---|---|---|---|
| 第二十九章 | 硬件 | 电气特性 ((A)级产品，目标系统) | X1振荡器特性 | 由于复位后CPU使用内部高速振荡时钟进行操作，因此用户要使用振荡稳定时间计数器的状态寄存器（OSTC）检测X1时钟振荡稳定时间。在充分评估了所使用的振荡器的振荡稳定时间后再确定OSTC和振荡稳定时间选择寄存器（OSTS）的振荡稳定时间。 | p. 594 □ |
| | | | XT1 荡器特性 | 在使用X1时钟时，上图中被虚线包围的部分的配线应按照如下配线方法配线，以防止连接线电容产生不利影响。<br>• 连接线越短越好。<br>• 连接线不应与其他信号线交叉。<br>• 流经的电流变化较大的信号线不要在振荡器周围布线。<br>• 要保持振荡器电容器的接地点电压与Vss相同。<br>• 不要将电容的地信号接入大电流地。<br>• 不要从振荡器获取信号。 | p. 595 □ |
| | | | | XT1振荡器是一个低振幅电路，用于降低功耗，但由于比X1振荡器更容易受到噪音干扰，因此在使用XT1时钟时应特别注意连线方式。 | p. 595 □ |
| 第三十一章 | 硬件 | 推荐焊接条件 | – | 不要一起使用不同的焊接方法（除部分加热法以外）。 | p. 615 □ |
| 第三十二章 | 软件 | 等待 | – | 当CPU使用副系统时钟而外围硬件时钟停止时，不要使用产生等待请求的方式访问上述寄存器。 | p. 617 □ |

# 附录 E 修订历史

## E.1 本版中主要修订之处

(1/6)

| 页码 | 描述 | 分类 |
|---|---|---|
| 整篇 | 添加(A)标准产品<br>PD78F0511GA(A)-GAM-AX, 78F0511GB(A)-GAF-AX, 78F0512GA(A)-GAM-AX, 78F0512GB(A)-GAF-AX, 78F0513GA(A)-GAM-AX, 78F0513GB(A)-GAF-AX, 78F0514GA(A)-GAM-AX, 78F0515GA(A)-GAM-AX | (d) |
| | 扩大电容的范围(“0.47 μF: target” → “0.47 to 1 μF: 建议) | (b) |
| | 修改电源电压斜率从“0.5 V/ms (MAX.)” 到“0.5 V/ms (MIN.)” | (b) |
| 引言 | | |
| p. 7, 8 | 添加如下的器件相关文档<br>• 78K0/Kx2 Flash 存储器编程(编程器)应用手册<br>• 78K0/Kx2 Flash 存储器自编程用户手册<br>• PG-FPL3 Flash 存储器编程器用户手册 | (e) |
| 第一章 概述 | | |
| p. 18 | 删除 **1.1 特征** 中关于产品过程分类管理的描述 | (d) |
| p. 19 | 修改 **1.3 订购信息** | (d) |
| pp. 25, 26 | 删除 **1.5 78K0/Kx2 系列产品**中关于产品过程分类管理的描述 | (d) |
| p. 29 | 删除 **1.7 功能概述**中关于产品过程分类管理的描述 | (d) |
| 第二章 引脚功能 | | |
| p. 36 | 在 **2.2.4 P30 ～ P33 (端口 3)** 中的添加**注意事项 2**，**注**，和**备注** 1 | (c) |
| p. 38 | 在 **2.2.8 P120 ～ P124 (端口 12)**中的添加**注意事项**，**注**，和**备注** 1 | (c) |
| pp. 40, 41 | 在 **表 2-2 引脚 I/O 电路类型**中添加如下内容<br>• 添加 P31/INTP2/OCD1A 引脚的**注 2** 和 **3** 和**备注**<br>• 添加 P121/X1/OCD0A 引脚的**注 2** 和 **5** 和**备注**<br>• 添加 $\overline{\text{FLMD0}}$ 引脚的**注 4**<br>• 添加 $\overline{\text{RESET}}$ 引脚未用时的连接 | (a), (c) |
| 第三章 **CPU** 结构 | | |
| pp. 45 ～ 51 | 在**图 3-1 存储器映射图(μPD78F0511)** ～ **3-7 存储器映射图(μPD78F0515D)**中添加**备注** 和 block 号图 | (c) |
| p. 52 | 添加**表 3-2 Flash 存储器的地址值和 block 号之间的关系** | (c) |
| 第四章 端口功能 | | |
| p. 94 | 修改**表 4-4 P20/ANI0～ P27/ANI7 引脚的设置功能**中的数字输入和输出的设置 | (a) |
| p. 94 | 在 **4.2.3 端口 2** 中添加**注意事项** | (c) |
| p. 95 | 在 **4.2.4 端口 3** 中添加**注意事项 2**, **注**和**备注 1** | (c) |
| p. 102 | 在 **4.2.8 端口 12** 中添加**注意事项 2**, **注**和**备注 1** | (c) |

**备注** 上表中的修改“分类”如下
(a): 错误纠正, (b): 添加/修改规范, (c): 添加/修改描述或者注释, (d): 添加/修改包,零件号,或管理分类,(e): 添加/修改相关文档

(2/6)

| 页码 | 描述 | 分类 |
|---|---|---|
| p. 108 | 在图 4-21 端口模式寄存器的格式中添加注意事项 | (c) |
| p. 109 | 在图 4-22 端口寄存器的格式中添加注 2 | (c) |
| p. 111 | 从图 4-24. A/D 端口配置寄存器 (ADPC) 的格式中删除注意事项 2 | (c) |
| p. 114 | 修改表 4-6 ANI0/P20 ～ ANI7/P27 引脚的设置功能中的数字输入和输出的设置 | (a) |
| p. 115 | 添加 4.6 端口寄存器n (Pn)的 1 位操作指令的注意事项 | (c) |
| 第五章　时钟发生器 | | |
| p. 118 | 在图 5-1 时钟发生器的框图中添加 “或”电路 | (a) |
| p. 120 | 修改图 5-2 时钟操作模式选择寄存器的格式 (OSCCTL) 中的注意事项 2 和 3 (有关停止提供给 CPU 时钟的时间的描述) | (b) |
| p. 128 | 在 5.4.1 X1 振荡器和 5.4.2 XT1 振荡器中添加外部时钟输入的描述 | (c) |
| p. 133 | 修改图 5-12 上电时的时钟发生器操作(设置 1.59 V POC 模式 (选项字节: POCMODE = 0))中的电压振荡稳定时间和复位处理时间以及添加注 1 关于等待振荡精确稳定 | (b), (c) |
| pp. 134, 135 | 在图 5-13 上电时的时钟发生器操作(设置 2.7V/1.59V POC 模式(选项字节: POCMODE = 1))中添加注意事项 1 和振荡精确稳定等待时间和修改复位处理时间。 | (c) |
| pp.136, 137 | 部分修改 5.6.1(1) <1>设置频率(OSCCTL 寄存器)和 5.6.1(2) <1>设置频率(OSCCTL 寄存器)的注(当 AMPH = 1 时的 CPU 时钟停止时间) | (b) |
| p. 144 | 修改图 5-14 CPU 时钟状态转移图 (设置 1.59 V POC 模式 (选项字节: POCMODE = 0))的备注 | (c) |
| p. 149 | 修改表 5-6 改变 CPU 时钟中的当 AMPH = 1 时的 CPU 时钟停止时间 | (b) |
| p. 150 | 修改表 5-7 CPU 时钟切换所需时间与主系统时钟周期分频因子的备注 2 | (a) |
| 第六章　16 位定时器/事件计数器 00 | | |
| p. 157 | 添加(iii) 当 CR000 或 CR010 用作比较寄存器时的设置范围 | (a), (c) |
| p. 160 | 部分修改图 6-5 16 位定时器模式控制寄存器 00 (TMC00)的格式中的第 3 位和第 2 位(TMC003, TMC002)的描述 | (c) |
| p. 170 | 修改图 6-17 方波输出操作的寄存器设置示例的(c) 16 位定时器输出控制寄存器 00 (TOC00) | (a) |
| p. 171 | 修改图 6-18 方波输出软件处理示例的时序图 | (a) |
| p. 173 | 修改图 6-20 外部事件计数器模式下寄存器设置示例的(c) 16 位定时器输出控制寄存器 00 (TOC00) | (a) |
| p. 174 | 修改图 6-21 外部事件计数器模式下的软件处理示例 | (a) |
| p. 190 | 修改图 6-35 自由运行定时器模式时序示例<br>(CR000: 比较寄存器, CR010: 捕捉寄存器) (修改当 CR000 = 0000H 时的图) | (c) |
| p. 198 | 修改图 6-41 PPG 输出操作的寄存器设置示例的注意事项 | (a) |
| p. 202 | 修改图 6-44 单脉冲输出操作的寄存器设置示例的注意事项 | (a) |

备注　上表中的修改“分类”如下

(a): 错误纠正, (b): 添加/修改规范, (c): 添加/修改描述或者注释, (d): 添加/修改包,零件号,或管理分类,(e): 添加/修改相关文档

(3/6)

| 页码 | 描述 | 分类 |
|---|---|---|
| p. 215 | 修改**表 6-3 16 位定时器/事件计数器 00 每个通道的限制条件**中作为外部事件计数器，PPG 输出和单脉冲输出的操作限制项 | (c) |
| **第七章　8 位定时器/事件计数器 50 和 51** | | |
| p. 225 | 修改**图 7-7 8 位定时器模式控制寄存器 50 (TMC50)的格式**和**图 7-8　8 位定时器模式控制寄存器 51 (TMC51) 的格式**的**注意事项** | (c) |
| p. 229 | 修改 **7.4.2 用作外部事件计数器**的设置<1>中 TMC5n 的设置值 | (a) |
| **第八章　8 位定时器 H0 和 H1** | | |
| p. 239 | 修改**图 8-3 8 位定时器 H 比较寄存器 0n（CMP0n）的格式**的**注意事项** | (c) |
| p. 239 | 在 **8.2 (2) 8 位定时器 H 比较寄存器 1n (CMP1n)**中添加部分描述 | (c) |
| pp. 242, 243 | 修改**图 8-5　8 位定时器 H 模式寄存器 0 （TMHMD0）的格式**和 **图 8-6　8 位定时器 H 模式寄存器 1（TMHMD1）的格式**的**注意事项 1** | (c) |
| p. 244 | 部分修改**图 8-7　8 位定时器 H 载波控制寄存器 1（TMCYC1）的格式**中 RMC1 和 NRZB1 位的描述以及添加**注意事项** | (c) |
| p. 247 | 修改**图 8-10 间隔定时器/方波输出操作时序**的 **(c) 当 CMP0n = 00H 时的操作** | (a) |
| p. 254 | 部分修改 **8.4.3 (2)载波输出控制**的 RMC1 和 NRZB1 位的描述 | (c) |
| **第十二章　A/D 转换器** | | |
| p. 286 | 删除**图 12-8. 模拟输入通道选择寄存器 (ADS)的格式**的**注意事项 3** | (a) |
| p. 287 | 删除**图 12-9.　A/D 端口配置寄存器(ADPC)的格式**的**注意事项 2** | (a) |
| p. 288 | 修改**表 12-3 ANI0/P20 ～ ANI7/P27 引脚功能的设置**的数字输入和输出设置 | (a) |
| **第十三章　串行接口 UART0** | | |
| p. 300 | 修改 **13.1 串行接口 UART0 的功能**的最大传输率 | (b) |
| p. 318 | 在**表 14-5　波特率发生器的设置数据**中添加当波特率为 312500 bps 和 625000 bps 时的设置数据 | (b), (c) |
| **第十四章 串行接口 UART6** | | |
| p. 321 | 修改 **14.1 串行接口 UART6 的功能**中的最大传输率 | (b) |
| p. 333 | 修改**图 14-9　波特率发生器控制寄存器 6 (BRGC6)的格式**中的输出时钟选择范围和**备注 2** | (b) |
| p. 352 | 部分修改 **14.4.3 (2) 串行时钟的产生**的描述 | (b) |
| p. 354 | 在**表 14-5 波特率发生器的设置数据**中添加当目标波特率为 625000 bps 时的设置数据和修改**备注 2** | (b), (c) |
| p. 356 | 在**表 14-6　最大/最小可允许的波特率误差**中添加分频比(k)为 4 时的误差 | (b) |
| **第十六章　串行接口 IIC0** | | |
| p. 389 | 部分修改**图 16-7 IIC 标志寄存器 0 (IICF0)的格式**中的 STCEN 位清零的条件 | (a) |
| p. 410 | 在 **16.5.16　通信操作**中添加**(1) 主设备操作 (单主设备系统)**, **(2) 主设备操作(多主设备系统)**和 **(3) 从设备操作**的描述 | (c) |
| p. 411 | 部分修改 **图 16-23 主设备操作(单主设备系统)** | (c) |
| p. 416 | 部分修改 **图 16-25 从设备操作流程图(1)** | (c) |

**备注**　上表中的修改"分类"如下

(a): 错误纠正, (b): 添加/修改规范, (c): 添加/修改描述或者注释, (d): 添加/修改包,零件号,或管理分类,(e): 添加/修改相关文档

(4/6)

| 页码 | 描述 | 分类 |
|---|---|---|
| 第十八章 中断功能 | | |
| p. 457 | 添加表 **18-1 中断源列表**的注 **4** | (c) |
| p. 459 | 添加表 **18-2 相应中断请求源的标志**的注 **4, 6 和 8～10** | (c) |
| 第二十章 待机功能 | | |
| p. 482 | 在图 **20-4 通过复位释放 HALT 模式**中添加振荡精确稳定时间和修改复位处理时间 | (c) |
| p. 484 | 在表 **20-3 STOP 模式中的操作状态**中添加 注 **1** 和修改**对串行接口 IIC0** 的描述 | (a), (c) |
| p. 485 | 修改 **20.2.2(1) STOP 模式设置及操作状态**的**注意事项 4** | (b), (c) |
| p. 485 | 修改图 **20-5 释放 STOP 模式时的操作时序** | (a), (c) |
| pp. 486, 487 | 修改图 **20-6 通过产生中断请求释放 STOP 模式** | (a), (c) |
| p. 488 | 在图 **20-7 由复位释放 STOP 模式**中添加振荡精确稳定时间和修改复位处理时间 | (c) |
| 第二十一章 复位功能 | | |
| pp. 491, 492 | 在图 **21-2 由 $\overline{\text{RESET}}$ 输入进行复位的时序～ 21-4 在 STOP 模式下通过 $\overline{\text{RESET}}$ 输入进行复位的时序**中添加振荡精确稳定时间，修改复位处理时间 | (a) |
| 第二十二章 上电清零电路 | | |
| pp. 498, 499 | 部分修改 **22.1 上电清零电路的功能** 和 **22.3 上电清零电路的操作**的描述 | (a) |
| p. 500 | 在图 **22-2 由上电清零电路和低电压检测电路产生内部复位信号的时序**的**(1) 1.59 V POC 模式(选项字节: POCMODE = 0)**中修改电压稳定等待时间和复位处理时间并且添加振荡精确稳定等待时间和**注 3** | (b), (c) |
| p. 501 | 在图 **22-2 由上电清零电路和低电压检测电路产生内部复位信号的时序**的**(2) 2.7 V/1.59 V POC 模式(选项字节: POCMODE = 1)**中修改复位处理时间并添加振荡精确稳定等待时间和**注意事项 2** | (b), (c) |
| 第二十三章 低电压检测电路 | | |
| p. 504 | 修改和添加 **23.1 低电压检测电路的功能**的描述 | (a), (c) |
| p. 506 | 修改图 **23-2 低电压检测寄存器(LVIM)的格式**中的 LVIMD 位的描述 | (a) |
| p. 508 | 修改和添加 **23.4 低电压检测电路的操作**的描述 | (a), (c) |
| pp. 514, 517 | 修改 **23.4.2 当用于中断**的描述 | (a), (c) |
| pp. 515, 516, 518 | 修改 **图 23-7 低电压检测电路中断信号的产生时序(检测供电电压($V_{DD}$)的等级)**和图 **23-8 低电压检测电路中断信号产生时序(检测外部输入引脚的输入电压(EXLVI)等级)** | (a) |
| p. 519 | 修改 **23.5 低电压检测电路的注意事项**中**(2) 用于中断**的描述 | (a), (c) |

备注　上表中的修改“分类”如下

(a): 错误纠正, (b): 添加/修改规范, (c): 添加/修改描述或者注释, (d): 添加/修改包,零件号,或管理分类,(e): 添加/修改相关文档

(5/6)

| 页码 | 描述 | 分类 |
|---|---|---|
| 第二十四章 选项字节 | | |
| p. 522 | 在 **24.1 选项字节的功能**中添加 **(1)0080H/1080H** 和**(2)0081H/1081H** 的注意事项 | (c) |
| 第二十五章 FLASH 存储器 | | |
| p. 530 | 修改**表 25-3 78K0/KC2 系列产品与专用 Flash 编程器之间的连线**的注 2 | (c) |
| p. 532, 534 | 在**图 25-4. UART (UART6) 模式下使用适配器对 Flash 存储器进行写操作示例(44 引脚产品)**和**图 25-6. UART (UART6) 模式下使用适配器对 Flash 存储器进行写操作示例(48 引脚产品)**中添加 注 | (c) |
| p. 536 | 在**图 25-9 专用 Flash 编程器的通信(UART6)**中添加 注 | (c) |
| p. 536 | 修改**表 25-4 引脚连接**的注 1 | (c) |
| p. 537 | 修改**图 25-10. FLMD0 引脚连接示例**和描述 | (a) |
| pp. 539, 540 | 在 **25.6.6 其他信号引脚**中添加描述，注意事项 3, 注和备注 | (c) |
| p. 540 | 添加 **25.6.7 供电电压**的描述 | (c) |
| p. 542 | 修改**表 25-7 通信模式** | (a) |
| p. 543 | 修改**表 25-8 Flash 存储器控制命令** | (c) |
| p. 544 | 部分修改和添加 **25.8 安全性设置**的描述 | (a), (c) |
| p. 545 | 修改**表 25-10 允许安全性功能和命令之间的关系** | (a), (c) |
| p. 545 | 修改**表 25-11 在每个编程模式下设置安全性** | (a), (c) |
| p. 546 | 添加 **25.9 当使用 PG-FP4 时每个命令的执行时间(参考值)** | (c) |
| p. 547 | 删除以前版本的 **25.10 通过自编程进行 Flash 存储器编程**中的 注意事项 5 | (c) |
| p. 548 | 修改**图 25-18 自编程流程(重写 Flash 存储器)** | (c) |
| pp. 549 ～ 552 | 添加**表 25-13 用于自编程例子库的处理时间和中断响应时间** | (c) |
| p. 553 | 部分修改**图 25-19 引导交换功能**的引导启动位置 | (a) |
| 第二十六章 片上调试功能(仅用于μPD78F0513D 和μPD78F0515D) | | |
| pp. 555, 556 | 在**图 26-1 QB-78K0MINI 与μPD78F0513D，μPD78F0515D 的连接示例(当使用 OCD0A/X1 和 OCD0B/X2 时)**和**图 26-2 QB-78K0MINI 与μPD78F0513D，μPD78F0515D 的连接示例(当使用 OCD1A/P31 和 OCD1B/P32 时)**的注中添加建议电阻 | (c) |
| p. 556 | 添加**图 26-3 通过片上调试进行自编程时的 FLMD0 引脚连接** | (c) |
| 第二十八章 电气特性(标准产品) | | |
| pp. 571, 572 | 修改 **额定最大值**的如下几项<br>• 输出电流, 高(添加 P20 ～ P27 和 P121 ～ P124 的值)<br>• 输出电流, 低 (添加 P20 ～ P27 和 P121 ～ P124 的值) | (b) |
| p. 574 | 在**内部振荡器特性**中添加当 RSTS = 0 ～ 8 MHz 内部振荡器 时的值 | (b) |
| p. 574 | 修改**内部振荡器特性**的 MIN. 值，条件是内部低速振荡时钟频率($f_{RL}$)的电压是 1.8 V ≤ $V_{DD}$ < 2.7 V | (b) |
| p. 575 | 添加 X1 振荡的振荡器和 XT1 振荡的振荡器以及振荡器参数 | (c) |

**备注** 上表中的修改“分类”如下

(a): 错误纠正, (b): 添加/修改规范, (c): 添加/修改描述或者注释, (d): 添加/修改包,零件号,或管理分类,(e): 添加/修改相关文档

(6/6)

| 页码 | 描述 | 分类 |
|---|---|---|
| pp. 578 ～ 581 | 修改 **DC 特性**的如下几项<br>• 输入电压, 高(修改 P60 ～ P63 的值)<br>• 输入电压, 低(修改 P60 ～ P62 的值)<br>• 输出电压, 低 (添加如下条件下的值: P60 ～ P63, 2.7 V ≤ $V_{DD}$ < 4.0 V, $I_{OL1}$ = 5.0 mA)<br>• 供电电流(修改操作模式和 HALT 模式下方波输入时的值，并且添加振荡器连接的值)。添加 STOP 模式下 $T_A$ = –40 ～ +70°C 的值。添加了**注 2**，修改**注 1 和 5。**<br>• A/D 转换器操作电流(添加转换器不操作时的值(比较器操作))。添加了**注 2。**<br>• LVI 操作电流 (修改 MAX. 值) | (b) |
| p. 579 | 添加以下 **DC 特性**<br>• 外部时钟输入电压 | (b) |
| pp. 582, 585 | 修改 **AC 特性**的如下几项<br>• **(1) 基本操作**的 TI000, TI010, TI001, TI011 输入高电平宽度, 低电平宽度(添加 1.8 V ≤ $V_{DD}$ < 2.7 V 条件下的值)<br>• **(2) 串行接口**的 **(a) UART6** 和 **(b) UART0** 的传输率(修改值) | (b) |
| p. 590 | 在 **LVI 电路特性**中添加外部输入引脚的检测电压的 MIN. 和 MAX. 值 | (b) |
| p. 591 | 在 **Flash 存储器编程特性**的**基本特性**中添加 **注 1** 和写时间的值。删除以前版本的“**(2) 串行写操作特性**” 和其他手册的介绍 | (b), (c) |
| **第二十九章 电气特性((A)级别产品,目标系统)** | | |
| p. 592 | 增加章 | (c) |
| **第三十章 封装图** | | |
| pp. 612, 614 | 添加(A) 级别产品的封装图 | (c) |
| **第三十一章 推荐焊接条件** | | |
| p. 615 | 增加章 | (c) |
| **附录 A 开发工具** | | |
| p. 622 | 在 **A.4 Flash 存储器写入工具**中添加 FP-LITE3, FA-78F0547GC-UBT-MX, FA-78F0547GK-8EU-MX ,**备注 2** | (d) |
| **附录 D 注意事项列表** | | |
| p. 634 | 增加章 | (c) |
| **附录 E 修订历史** | | |
| p. 665 | **添加了 E.2 前版的修订历史** | (c) |

**备注** 上表中的修改“分类”如下

(a): 错误纠正, (b): 添加/修改规范, (c): 添加/修改描述或者注释, (d): 添加/修改包,零件号,或管理分类,(e): 添加/修改相关文档

<R>

## E.2 以前版本的修订历史

如下是以前版本的修改历史。章节表示各个版本的章节。

(1/7)

| 版本 | 描述 | 章节 |
|---|---|---|
| 第二版 | 在 **1.1 特性**中添加具有片上调试功能的产品的**注** 和修改操作环境温度 | **第一章 概述** |
| | 在 **1.2 应用**中添加支持汽车装备的特殊级别产品 | |
| | 修改 **1.3 订购信息** | |
| | 在 **1.4 引脚配置(顶视图)**里添加了**注**和对**注意事项 1** 的修改 | |
| | 修改 **1.5 78K0/Kx2 系列**中如下几项<br>•内部低速振荡时钟的供电电压范围<br>•POC 的检测电压<br>•操作环境温度 | |
| | 在 **1.6 框图** 中添加"片上调试"的引脚 | |
| | 修改 **1.7 功能概述**中的以下几项<br>• 高速系统时钟的振荡频率范围<br>• 内部低速振荡时钟的供电电压范围<br>• 操作环境温度<br>• 封装 | |
| | 修改 **1.7 功能概述**中定时器的概述 | |
| | 修改**表 2-1 引脚 I/O 缓冲器供电电源** | 第二章 引脚功能 |
| | 在 **2.1 引脚功能列表**中添加**注** | |
| | 修改**.2.2.11 AV<sub>REF</sub>** 的描述 | |
| | 在 **2.2.14 REGC** 中添加**注意事项** | |
| | 修改**表 2-2 引脚 I/O 电路类型**中引脚 P121/X1，P122/X2/EXCLK，P123/XT1 和 P124/XT2/EXCLKS 未使用时的建议连接方法 | |
| | 修改**表 3-1 设置内部存储器容量切换寄存器(IMS)和内部扩展 RAM 容量切换寄存器(IXS)的值** | 第三章 **CPU** 结构 |
| | 修改 **图 3-1 存储器映射图(μPD78F0511) ～ 图 3-7 存储器映射图(μPD78F0515D)** | |
| | 修改 **3.1.1** 中**(3)** 选项字节区域 和**(5)** 片上调试安全性 **ID** 设置区域**(**仅用于**μPD78F0515D)**的描述 | |
| | 修改 **3.4.4** 短直接寻址的 **[描述示例]** | |
| | 修改**表 4-1** 引脚 **I/O** 缓冲器电源 | 第四章 端口功能 |
| | 修改图 **4-2 P00** 的框图 | |
| | 修改图 **4-3 P01** 的框图 | |
| | 在 **4.2.2** 端口 **1** 中添加**注意事项** | |
| | 添加 **4.2.3** 端口 **2** 的描述并且添加**表 4-4 P20/ANI0 ～ P27/ANI7** 引脚的设置功能 | |
| | 在 **4.2.8** 端口 **12** 中添加**备注**以及修改**注意事项** | |
| | 修改图 **4-18 P120** 的框图 | |
| | 修改图 **4-19 P121～ P124** 的框图 | |

(2/7)

| 版本 | 描述 | 章节 |
|---|---|---|
| 第二版 | 在 **4.2.9 端口 13(仅 48 引脚产品)的备注**中添加图 | **第四章 端口功能** |
| | 在 **4.3 控制端口功能的寄存器**中添加**(4) A/D 端口配置寄存器 (ADPC)** | |
| | 在**表 4-5 使用复用功能时，端口模式寄存器和输出锁存器的设置 (2/2)**中添加**备注 2** 和**注 1** 和 **2** | |
| | 修改 **5.1 (1) 主系统时钟**中 X1 振荡器的振荡频率范围外部主系统时钟 | **第五章 时钟发生器** |
| | 添加 **5.1 (3) 内部低速振荡时钟**的描述 | |
| | 修改**图 5-1 时钟发生器的框图** | |
| | 修改**图 5-3 处理器时钟控制寄存器(PCC)的格式** | |
| | 添加 **5.3 (3) 副系统时钟引脚的操作模式的设置** | |
| | 修改 **5.3 (8) 振荡稳定时间选择寄存器(OSTS)** | |
| | 修改 **5.4.1 X1 振荡器**的振荡频率范围 | |
| | 修改 **5.4.3 当复系统时钟没用时**的描述 | |
| | 添加**图 5-12 当上电时的时钟发生器操作(设置 1.59 V POC 模式  (选项字节: POCMODE = 0))** | |
| | 添加**图 5-13** 上电时的时钟发生器操作**(设置 2.7V/1.59V POC 模式(**选项字节**: POCMODE = 1))** | |
| | 修改 **5.6.1 高速系统时钟控制** | |
| | 修改 **5.6.2 控制内部高速振荡时钟示例** | |
| | 修改 **5.6.3 控制副系统时钟示例** | |
| | 修改**表 6-4 CPU 和外部硬件采用的时钟及寄存器设置**的描述 | |
| | 在**图 5-14 CPU 时钟状态转移图 (设置 1.59 V POC 模式 (选项字节: POCMODE = 0))**中添加**备注** | |
| | 修改**表 5-5 CPU 时钟切换与 SFR 寄存器设置示例**中的以下几项<br>**(3) 复位释放后(A)CPU 使用副系统时钟(D)**<br>**(4) CPU 时钟从内部高速振荡时钟(B)切换到高速系统时钟(C)**<br>**(5) CPU 时钟从内部高速振荡时钟(B)切换到副系统时钟(D)**<br>**(7) CPU 时钟从高速系统时钟(C)切换到副系统时钟(D)**<br>**(9) CPU 时钟从副系统时钟(D)切换到高速系统时钟(C)** | |
| | 修改**表 5-6 切换 CPU 时钟** | |
| | 添加 **5.6.8 CPU 时钟和主系统时钟切换所需的时间** | |
| | 添加 **5.6.9 时钟振荡停止前的状况** | |
| | 添加 **5.6.10 外部硬件与源时钟** | |
| | 修改章节 | **第六章 16 位定时器/事件计数器 00** |
| | 修改 **7.2 (2) 8-位定时器比较寄存器 5n (CR5n)** | **第七章 8 位定时器/事件计数器 50 和 51** |

(3/7)

| 版本 | 描述 | 章节 |
|---|---|---|
| 第二版 | 修改图 8-2 8 位定时器 H1 的框图 | 第八章 8 位定时器 H0 和 H1 |
| | 修改 8.2 中的(1)8 位定时器 H 比较寄存器 0n (CMP0n) 和 (2)8-位定时器 H 比较寄存器 1n (CMP1n)的描述 | |
| | 修改图 8-6 8 位定时器 H 模式寄存器 1 (TMHMD1)的格式 | |
| | 修改图 8-12 (e) 修改 CMP1n 的时序 （CMP1n = 02H → 03H， CMP0n = A5H） | |
| | 修改 8.4.3 载波发生器操作（仅用于 8 位定时器 H1）的描述 | |
| | 在图 8-13 发送时序中添加<3> | |
| | 在 8.4.3 的设置中添加 <8> | |
| | 修改图 8-15 中的(a) 当 CMP01 = N, CMP11 = N 时的操作和(b) 当 CMP01 = N， CMP11 = M 时的操作 | |
| | 修改 图 8-15 (c) 当 CMP11 改变时的操作的描述 | |
| | 修改 10.1 看门狗定时器的功能的描述 | 第十章 看门狗定时器 |
| | 在 10.4.1 控制看门狗定时器的操作中添加的描述并且添加注意事项 4 | |
| | 在图 11-2 时钟输出选择寄存器(CKS)的格式中添加注 1 和 注意事项 1 和 2。 | 第十一章 时钟输出控制器 (仅 48 引脚产品) |
| | 修改 12.2 A/D 转换器的配置中的以下几项<br>(2) 采样& 保持电路<br>(3) 串联电阻串<br>(5) 逐次逼近寄存器 (SAR)<br>(9) $AV_{REF}$ 引脚 | 第十二章 A/D 转换器 |
| | 在表 12-2 A/D 转换时间选择中添加注意事项 1 和 4 | |
| | 修改图 12-8 模拟输入通道选择寄存器(ADS)的格式中的注意事项 2 和 3 | |
| | 修改 12.3 (5) A/D 端口配置寄存器(ADPC)的描述 | |
| | 修改图 12-9 A/D 端口配置寄存器(ADPC)的格式中的注意事项 1 和 2 | |
| | 修改表 12-3 ANI0/P20 ～ ANI7/P27 引脚的设置功能 | |
| | 修改 12.4.1 A/D 转换器的基本操作 | |
| | 修改图 12-11 A/D 转换器的基本操作的描述 | |
| | 修改 12.4.2 输入电压和转换结果的表达 | |
| | 修改 12.4.3 A/D 转换器操作模式的描述 | |
| | 修改 12.6 A/D 转换器使用注意事项中的以下几项的描述<br>(1) STOP 模式中的操作电流<br>(4) 解决噪音问题的方法<br>(6) ANI0 ～ ANI7 引脚的输入阻抗<br>(11) 内部等效电路 | |

(4/7)

| 版本 | 描述 | 章节 |
|---|---|---|
| 第二版 | 在 **13.1 (2) 异步串行接口(UART)模式**中添加最大传输率和**注意事项 4** | **第十三章 串行接口 UART0** |
| | 在 **13.2 (3) 发送移位寄存器 0 (TXS0)**中添加**注意事项 1** | |
| | 在图 **13-2 异步串行接口操作模式寄存器** 0 **(ASIM0)的格式**中添加**注意事项 5** | |
| | 修改 **13.3 (2) 异步串行接口接收错误状态寄存器 0 (ASIS0)**的描述 | |
| | 修改图 **13-9 接收完成中断请求时序**中的**注意事项 1** | |
| | 添加表 **13-4 TPS01 和 TPS00 的设置值** | |
| | 在 **14.1 (2) 异步串行接口(UART)模式**中添加传输率和 **注意事项 4** 和 **5** | **第十四章 串行接口 UART6** |
| | 修改图 **14-1 LIN 发送操作** | |
| | 修改图 **14-2 LIN 接收操作** | |
| | 在 **14.2 (3) 发送缓冲寄存器 6 (TXB6)**中添加**注意事项 3** | |
| | 在图 **14-5 异步串行接口操作模式寄存器 6 (ASIM6)的格式**中添加**注意事项 4** 和 **5** | |
| | 修改 **14.3 (2) 异步串行接口接收错误状态寄存器 6 (ASIS6)**的描述 | |
| | 在图 **14-10 异步串行接口控制寄存器 6 (ASICL6) 的格式**中添加**注意事项 6** | |
| | 修改 **14.3 (7) 输入切换控制寄存器(ISC)**的描述 | |
| | 修改 **14.4.2 (2) (e) 正常接收**中的**注意事项 1** | |
| | 修改图 **15-1 串行接口 CSI10 的框图** | **第十五章 串行接口 CSI10** |
| | 修改图 **15-2 串行操作模式寄存器 10（CSIM10）的格式**的**注 2** | |
| | 修改图 **15-3 串行时钟选择寄存器 10（CSIC10）的格式**的**注意事项 2** | |
| | 修改 **15.4.1 (1) 使用的寄存器**中的 CSIM10 的**注 1** | |
| | 在图 **15-7 第 1 位的输出操作**中添加**(b) 类型 3: CKP1n0= 1, DAP1n0= 0** 和**(d) 类型 4: CKP10 = 1, DAP10 = 1** | |
| | 在图 **15-8 SO1n 引脚的输出值（最后一位）**中添加**(b) 类型 3: CKP10 = 1, DAP10 = 0** 和 **(d) 类型 4: CKP10 = 1, DAP10 = 1** | |
| | 修改图 **16-1 串行接口 IIC0 的框图** | |
| | 在 **16.2 (1) IIC移位寄存器 0 (IIC0)**中添加**注意事项 2** 并且添加**(2) 从地址寄存器 0 (SVA0)**的描述 | |
| | 添加 **16.2 (13) 停止条件发生器** | |
| | 在图 **16-5 IIC控制寄存器 0 (IICC0)的格式 (1/4)**中添加IICE0 的描述以及添加**注意事项** | |

(5/7)

| 版本 | 描述 | 章节 |
|---|---|---|
| 第二版 | 在图 **16-5 IIC 控制寄存器 0 (IICC0)的格式 (2/4)**中添加注 **2** | **第十六章 串行接口 IIC0** |
| | 在图 **16-5 IIC 控制寄存器 0 (IICC0)的格式(3/4)** 中添加 STT0 的描述 | |
| | 在图 **16-7 IIC 标志寄存器 0 (IICF0) 的格式**中添加 STCF 和 IICBSY 的清零条件 | |
| | 修改**16.3 (4) IIC时钟选择寄存器 0 (IICCL0)**的描述 | |
| | 修改 **16.3 (6) I²C 传送时钟设置方法**的描述 | |
| | 修改**表 16-2 选择时钟设置** | |
| | 在 **16.5.4 应答信号 (ACK)**中添加不返回 ACK 的原因 | |
| | 添加**16.5.7 取消等待** | |
| | 修改**表 16-6 等待周期** 和 **图 16-20 通信预约时序** | |
| | 修改**表 16-7 等待周期** | |
| | 在 **16.5.15 其他注意事项**中添加 (4) ～ (6) | |
| | 修改 **16.5.16 (1) 主设备操作 (单主设备系统)** 和 **(2) 主设备操作(多主设备系统)** | |
| | 修改**图 16-25 从设备操作流程图(1)** 和 **图 16-26 从设备操作流程图(2)** | |
| | 在 **16.5.17 (1) 主设备操作**中添加**(a) (i)** 当 **WTIM0 = 0** 的注并且修改 **(ii)** 当 **WTIM0 = 1** | |
| | 在 **16.5.17 (1) 主设备操作**的**(b) (i)** 当 **WTIM0 = 0** 中添加注 **1**～ **3** | |
| | 在 **16.5.17 (1) 主设备操作**中添加**(c) (i)** 当 **WTIM0 = 0** 的 注 | |
| | 修改 **16.5.17** 中如下条件下的 IICS0 寄存器的值<br>**(2) (d) (i)** 当 WTIM0 = 0 (重启后，地址不相等 (= 非扩展码))<br>**(2) (d) (ii)** 当 WTIM0 = 1 (重启后，地址不相等 (= 非扩展码))<br>**(3) (d) (i)** 当 WTIM0 = 0 (重启后，地址不相等 (= 非扩展码))<br>**(3) (d) (ii)** 当 WTIM0 = 1 (重启后，地址不相等 (= 非扩展码))<br>**(6) (d) (ii)** 扩展码<br>**(6) (e)** 数据传送期间由于停止条件导致仲裁失败时<br>**(6) (h) (ii)** 当 WTIM0 = 1 | |
| | 添加 **16.5.17 (5) 发生仲裁失败时的操作 (仲裁失败后作为从设备操作)**和 **(6) 发生仲裁失败时的操作 (仲裁失败后不进行通信)**的描述 | |
| | 在 **16.5.17 (6) 发生仲裁失败时的操作 (仲裁失败后不进行通信)**中添加**(i) 当 WTIM0 = 0** 时的如下条件<br>**(f) 要产生重启条件时，由于低电平数据导致仲裁失败时**<br>**(g) 要产生重启条件时，由于停止条件导致仲裁失败时**<br>**(h) 要产生停止条件时，由于低电平数据导致仲裁失败时** | |
| | 修改**图 16-27 主设备至从设备的通信举例(当主设备和从设备都选择 9-时钟等待时)** 和**图 16-28 从设备至主设备通信举例(当主设备选择 8-时钟等待，从设备选择 9-时钟等待时)** | |

(6/7)

| 版本 | 描述 | 章节 r |
|---|---|---|
| 第二版 | 修改图 17-7 除法操作的时序图(DCBA2586H ÷ 0018H) | 第十七章 乘法器/除法器 (仅用于 μPD78F0514, 78F0515 和 78F0515D) |
| | 修改 20.1.1 待机功能的注意事项 3 | 第二十章 待机功能 |
| | 修改 20.1.2 (2) 振荡稳定时间选择寄存器(OSTS)的描述 | |
| | 在表 20-1 HALT 模式下的操作状态中添加时钟输出和蜂鸣器输出项以及添加注 | |
| | 修改图 20-4 由复位释放 HALT 模式 | |
| | 在表 20-3 STOP 模式中的操作状态中添加时钟输出和蜂鸣器输出项并且添加注 | |
| | 修改图 20-5 STOP 模式释放时的操作时序 | |
| | 修改图 20-7 由复位释放 STOP 模式 | |
| | 修改图 21-2 由 RESET 引脚输入进行复位的时序 | 第二十一章 复位功能 |
| | 修改图 21-3 由看门狗定时器溢出进行复位的时序 | |
| | 修改图 21-4 在 STOP 模式下通过 RESET 输入进行复位的时序 | |
| | 在表 21-1 复位期间的操作状态 | |
| | 修改表 21-2 复位响应后的硬件状态(3/3)的注释 | |
| | 在 22.1 上电清零电路的功能中添加 2.7 V/1.59 V POC 模式的描述 | 第二十二章 上电清零电路 |
| | 修改 22.3 上电清零电路的操作 | |
| | 修改图 22-3 复位释放后软件处理过程示例(1/2) | |
| | 修改图 23-1 低电压检测电路的框图 | 第二十三章 低电压检测电路 |
| | 修改图 23-3.低电压检测等级选择寄存器（LVIS）的格式 | |
| | 在图 23-5 低电压检测电路内部复位信号的产生时序(检测供电电压($V_{DD}$)等级)中添加 (2) 2.7 V/1.59 V POC 模式(选项字节: POCMODE = 1) | |
| | 修改图 23-7 低电压检测电路中断信号的产生时序(检测供电电压($V_{DD}$)的等级)中的(1)1.59 V POC 模式(选项字节: POCMODE = 0)以及增加(2) 2.7 V/1.59 V POC 模式 (option byte: POCMODE = 1) | |
| | 修改 图 23-8 低电压检测电路中断信号产生时序(检测外部输入引脚的输入电压(EXLVI)等级) | |
| | 修改 图 23-9 复位释放后软件处理过程示例(1/2) | |
| | 修改 24.1 选项字节的功能 的描述 | 第二十四章 选项字节 |
| | 在图 24-1 选项字节的格式中修改注释和添加 0081H/1081H ～ 0084H/1084H 区域的设置 | |
| | 修改 设置选项字节的软件示例的描述 | |

(7/7)

| 版本 | 描述 | 章节 |
| --- | --- | --- |
| 第二版 | 在图 **25-1 内部存储器容量切换寄存器(IMS)的格式**中添加**注意事项** | **第二十五章 FLASH 存储器** |
| | 在图 **25-2 内部扩展 RAM 容量切换寄存器 (IXS)的格式**中添加**注意事项** | |
| | 修改图 **25-3 3 线串行 I/O (CSI10) 模式下使用适配器对 Flash 存储器进行写操作示例(44 引脚产品)**中 $V_{DD}$ 的值 | |
| | 修改图 **25-4 UART(UART6) 模式下使用适配器对 Flash 存储器进行写操作示例(44 引脚产品)** 中 $V_{DD}$ 的值 | |
| | 修改图 **25-5 3 线串行 I/O (CSI10) 模式下使用适配器对 Flash 存储器进行写操作示例(48 引脚产品)**中 $V_{DD}$ 的值 | |
| | 修改图 **25-6 UART(UART6) 模式下使用适配器对 Flash 存储器进行写操作示例(48 引脚产品)** 中 $V_{DD}$ 的值 | |
| | 修改 **25.5 的(1) CSI10** 和 **(2) UART6** 中传输率 | |
| | 修改**表 25-7 通信模式**的速度栏中的传输率 | |
| | 添加 **25.8 安全性设置** | |
| | 修改 **25.9.1 引导交换功能** | |
| | 修改章节 | **第二十六章 片上调试功能(仅用于 μPD78F0513D 和 μPD78F0515D)** |
| | 修改章节 | **第二十八章 电气特性(目标)** |
| | 添加封装图 | **第二十九章 封装图** |
| | 修改**表 30-1 发出等待请求的寄存器和 CPU 等待的时钟数** 中的 A/D 转换器的<最大/最小等待时钟个数的条件> | **第三十章 等待的注意事项** |
| | 在图 **A-1 开发工具配置**中添加**(2)当使用片上调试仿真器 QB-78K0MINI 时** | **附录 A 开发工具** |
| | 添加 **A.5.2 当使用片上调试仿真器 QB-78K0MINI 时** | |
| | 增加章 | **附录 B 目标系统设计的注意事项** |
| | 增加章 | **附录 D 修改历史** |

*For further information,*
*please contact:*

**NEC Electronics Corporation**
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668,
Japan
Tel: 044-435-5111
http://www.necel.com/

**[America]**

**NEC Electronics America, Inc.**
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
http://www.am.necel.com/

**[Europe]**

**NEC Electronics (Europe) GmbH**
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
http://www.eu.necel.com/

**Hanover Office**
Podbielski Strasse 166 B
30177 Hanover
Tel: 0 511 33 40 2-0

**Munich Office**
Werner-Eckert-Strasse 9
81829 München
Tel: 0 89 92 10 03-0

**Stuttgart Office**
Industriestrasse 3
70565 Stuttgart
Tel: 0 711 99 01 0-0

**United Kingdom Branch**
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908-691-133

**Succursale Française**
9, rue Paul Dautier, B.P. 52180
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

**Sucursal en España**
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

**Tyskland Filial**
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 638 72 00

**Filiale Italiana**
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02-667541

**Branch The Netherlands**
Limburglaan 5
5616 HR Eindhoven
The Netherlands
Tel: 040 265 40 10

**[Asia & Oceania]**

**NEC Electronics (China) Co., Ltd**
7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian
District, Beijing 100083, P.R.China
TEL: 010-8235-1155
http://www.cn.necel.com/

**NEC Electronics Shanghai Ltd.**
Room 2509-2510, Bank of China Tower,
200 Yincheng Road Central,
Pudong New Area, Shanghai P.R. China P.C:200120
Tel: 021-5888-5400
http://www.cn.necel.com/

**NEC Electronics Hong Kong Ltd.**
12/F., Cityplaza 4,
12 Taikoo Wan Road, Hong Kong
Tel: 2886-9318
http://www.hk.necel.com/

**Seoul Branch**
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737

**NEC Electronics Taiwan Ltd.**
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-2719-2377

**NEC Electronics Singapore Pte. Ltd.**
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
http://www.sg.necel.com/

**G05.12A**